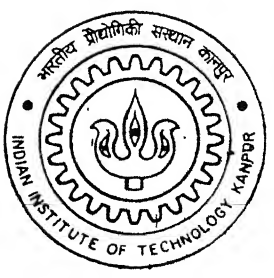
CONTROL AND PATH-PLANNING OF PARALLEL AND HYBRID MANIPULATORS

By

SHAMIK SEN



DEPARTMENT OF MECHANICAL ENGINEERING

Indian Institute of Technology Kanpur

JANUARY, 2002

CONTROL AND PATH-PLANNING OF PARALLEL AND HYBRID MANIPULATORS

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
Master of Technology

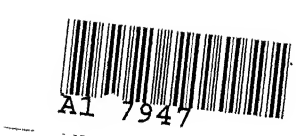
by

SHAMIK SEN

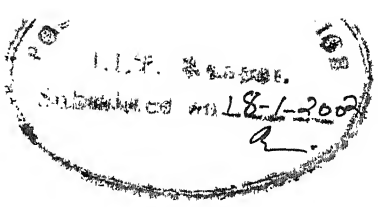
to the

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
INDIA

January, 2002



CERTIFICATE



It is certified that the work contained in the thesis entitled "***Control and Path-Planning of Parallel and Hybrid Manipulators***", by ***Mr. Shamik Sen*** , has been carried out under our supervision and that this work has not been submitted elsewhere for a degree.

Dr. A.K. Mallik
Professor
Dept. of Mechanical Engineering and
I.I.T Kanpur

Dr. B. Dasgupta
Asst. Professor
Dept. of Mechanical Engineering
I.I.T. Kanpur

January, 2002

Acknowledgements

I express my sincere gratitude, regards and thanks to my supervisors Prof. B. Dasgupta and Prof. A. K. Mallik for their excellent guidance, invaluable suggestions and generous help at all the stages of my research work. Their interest and confidence in me was the reason for all the success I have made.

I am extremely grateful to my friends Pradipta, Shamik, Seresta, Abir, Arnab, Arijit, Prithwijit, Suman, and Debadi for all the encouragement and support I have received from them at all times. I am glad to record my thanks to my Robotics Lab mates Kaushik, Pawan, Ankur and Gaurav whose kindness and cooperation I remember with gratitude.

Once again, I would like to thank all my classmates for their smile and friendship making the life at I.I.T. Kanpur enjoyable and memorable.

Shamik Sen

Dedicated to My Parents and Teachers

Abstract

This thesis deals with the topics of control and path-planning of constrained mechanical systems such as parallel and hybrid manipulators. Such systems differ from the more common serial manipulators in having more than one loop with one or more passive joints, resulting in more complex kinematics, and highly nonlinear equations describing their dynamics.

Three different control strategies, namely, PD control, Model-based control and Optimal control have been implemented for controlling some parallel and hybrid manipulators. The characteristic features of each of these strategies have been discussed, and a qualitative performance evaluation has been done through different examples applied to individual manipulators.

The problem of singularity-free path-planning has been posed for parallel manipulators. The Variational Approach has been discussed and applied for singularity-free path-planning within the workspace. The above procedure has been applied for a number of different cases.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

Parallel manipulators are robotic devices which differ from the traditional serial manipulators by virtue of their kinematic structure. Parallel manipulators are composed of multiple closed kinematic loops. These loops are formed by two or more kinematic chains that connect a moving platform to a base, where one joint in each chain is actuated and the other joints are passive. This kinematic structure allows them to be driven by actuators positioned at or near the base of the manipulator.

In contrast, serial manipulators do not have closed kinematic loops and are usually actuated at each joint along the serial linkage. Accordingly, the actuators located at each joint along the serial linkage account for a significant portion of the loading experienced by the manipulator, whereas the links of a parallel manipulator generally need not carry the load of the actuators. This allows the parallel manipulator links to be made lighter than the links of an analogous serial manipulator, and make them more suitable for various high-speed operations.

The advantages of a serial manipulator are large workspace and dextrous manoeuvreability like the human arm. But due to their cantilever structure, serial manipulators suffer from some drawbacks like lower operational accuracy and lower load carrying capacity. These disadvantages can be taken care of by designing manipulators with closed kinematic loops. This is the reason why research interest has shifted considerably from serial manipulators to parallel manipulators. A new type of manipulator architecture can be developed by combining the characteristics of serial and parallel manipulators. Such manipulators, called hybrid manipulators, retain the advantages of both classes to some extent. A hybrid manipulator structure can be developed by an in-parallel combination of serially actuated chains or by an in-series combination of parallel-actuated modules or by a more complicated series-parallel combination.

1.2 Comparison between Serial Manipulators and Parallel Manipulators

The differences between a serial manipulator and a parallel manipulator are as follows:

1. For a serial manipulator, the degrees-of-freedom (DOF) of the system is given by

$$DOF = \sum_{i=1}^{j} f_i$$

where f_i corresponds to the DOF of the i-th joint of the manipulator consisting of j joints. In general for serial manipulators, the DOF is equal to the number of links as each link normally has single degree of freedom. For parallel manipulators, the expression for DOF is given by

$$DOF = m(n - j - 1) + \sum_{i=1}^{j} f_i$$

where n is the number of links (including the fixed base link) and j is the number of joints, respectively. $m = 3$ for planar case and $m = 6$ for spatial case.

2. The problem of direct kinematics is to find the position and orientation of the output link, given the input joint variables. The direct kinematics of serial manipulators is very simple and can be determined by using the Denavit-Hartenberg parameters and homogeneous coordinate transformation. On the other hand, direct kinematics of parallel manipulators involve the simultaneous solution of a system of nonlinear equations. The direct kinematics of serial manipulators is straightforward and *unique*, while that of parallel manipulators is not unique. In general, the direct kinematics problem of a hybrid or a parallel manipulator is more complicated than that of a serial manipulator and the kinematics of parallel and hybrid manipulators is a continuing subject of research.

3. The problem of inverse kinematics is to find out the required values of the input joint variables for a given position and orientation of the output member. The inverse kinematics of serial manipulators invloves the solution of high order polynomial equations and results in multiple solutions. Inverse kinematics of serial manipulators is not very trivial. In comparison to serial manipulators, inverse kinematics of parallel manipulators is straightforward and *unique*.

4. The instantaneous kinematics, and its dual the statics, of serial manipulators, involves the Jacobian matrix relating the joint space variables (joint rates or joint torques) to the Cartesian variables (end-effector velocities or forces and moments). In case of parallel manipulators, the analogous quantity is the force transformation matrix. At singular

configurations in a serial manipulator, the Jacobian loses rank. In case of parallel manipulators, at a singular configuration, the force transformation matrix becomes rank deficient and it corresponds to a condition of force singularity where it gains one or more degrees of freedom.

5. The singularities of serial manipulators can be found out easily from the inverse kinematics - these correspond to either completely folded configuration or completely extended configuration. At these configurations, the serial manipulator loses one or more degrees of freedom and at such locations joint rates go to infinity. The singularities of a parallel manipulator can be obtained from the singularity of the force transformation matrix. It is known that a serial manipulator can only *lose* degrees of freedom at a singularity, however, a parallel manipulator can *gain* or *lose* one or more degrees of freedom at a singularity. For parallel manipulators with prismatic actuations, the loss of one or more degree of freedom corresponds to positions where the leg length of the manipulator becomes zero or the mechanism encounters either an internal or external workspace boundary.

6. The dynamics of serial manipulators is given by ordinary differential equations. These equations can be obtained by a variety of methods such as Newton-Euler or Euler-Lagrange procedure. The dynamic equations of motion of parallel manipulators can be expressed by ordinary differential equations or by a combination of differential and algebraic equations. While it is widely accepted that the Euler-Lagrange formulation is better suited for deriving the closed-form dynamic equations of a serial manipulator, for parallel manipulators the Newton-Euler formulation is more suitable.

7. The control of serial manipulators can be easily done by simple control schemes such as independent joint proportional plus derivative (PD) control. In general, serial manipulators can be shown to be *controllable* except at singularities. For general parallel manipulators, control is much more difficult, and controllability studies have to be done to determine the controllability of the manipulators.

8. The kinematic redundancies in a serial manipulator can be introduced by adding extra links and joints. In parallel manipulators, force redundancy is introduced by adding extra limbs and actuated joints in parallel. Redundant serial manipulators find use in complex workspaces where serpentine-like motions of the manipulator are required for avoiding obstacles. Redundancies are used to reduce the singularities of parallel manipulators. However, adding redundant links to a parallel manipulator reduces the workspace of the manipulator.

1.3 Literature Survey

In the field of parallel manipulators, a lot of research has been done in the fields of direct and inverse kinematics, workspaces, and singularity analysis. Kim, Byun and Cho [1] have developed closed-form forward position solution for a 6-dof 3-PPSP parallel mechanism. Gregario and Castelli [2] have worked on the direct and inverse position analysis of a 3-PSP mechanism. Dhingra, Almadi and Kohli [3] have used Grobner-Sylvester hybrid method for displacement analysis of planar and spatial mechanisms. Fang and Huang [4] have solved the kinematics of a 3-dof parallel manipulator. Dunlop and Jones [5] have also worked on the position analysis of a 3-dof parallel manipulator. Martinez and Duffy [6] have derived expressions for the forward and inverse acceleration analysis of parallel manipulators. Tsai [7] used the principle of virtual work for solving the inverse dynamics of a Stewart-Gough manipulator. Dasgupta and Mruthyunjaya [8] have developed the closed-form dynamic equations of the Stewart platform manipulator. They have also solved the inverse dynamics of the Stewart platform manipulator. Dasgupta and Choudhury [9] have developed a general strategy based on the Newton-Euler formulation for closed-form dynamic formulation of parallel manipulators. Basu and Ghosal [10] have worked on singularity analysis of platform-type multi-loop spatial mechanisms. Gao, Liu and Chen [11] showed that the shapes of workspaces and the link-lengths are related in case of a 3-dof symmetric planar parallel manipulators. Notash [12] investigated the uncertainty configurations of three-branch parallel manipulators. Matone and Roth [13] studied the effects of actuation schemes and their effects on singular postures. Wang and Gosselin [14] have determined singularity loci of spatial 4-dof parallel manipulators. Ricard and Gosselin [15] have developed a new method for determination of workspaces of complex planar manipulators. Ghosal and Ravani [16] developed a differential-geometric approach for analyzing task-space point trajectories of serial and parallel manipulators. Bonev and Ryu [17] developed a geometrical method for computing the constant-orientation workspace of 6-PRRS parallel manipulators. While there have been a lot of research work done in the field of parallel manipulators on topics like position kinematics, dynamics and singularity analysis, research strictly addressing the problem of control of parallel manipulators has been rather scarce. Pfreundschuh, Kumar and Sugar [18] have worked on the design and control of a 3 DOF parallel manipulator. Cleary and Arai [19] have worked on construction and control of a prototype parallel manipulator. Nenchev and Uchiyama [20] have worked on singularity-consistent control of parallel manipulators. Though there have been a lot of research work done in the field of path-planning of serial manipulators, research in the field of path-planning of parallel manipulators is much less. Bhattacharya *et al.* [21] have developed a scheme for avoiding singularities of a Stewart platform by restructuring a pre-planned path in the vicinity of a singularity. Dasgupta and Mruthyunjaya [22] have developed an algorithm for planning a well-conditioned path between two end-points for a Stewart platform manipulator. The algorithm also indicates the non-existence of a valid path. Innocenti and Castelli [23] showed

that a singularity-free configuration change is possible for parallel manipulators. Liu, Jin and Gao [24] have worked on optimum design of 3-dof spherical parallel manipulators with reference to the conditioning and stiffness indices. Lee, Duffy and Keller [25] have developed an index for checking the stability of parallel manipulators. Boudreau and Gosselin [26] have used genetic algorithms for synthesis of planar parallel manipulators. Tsai and Joshi [27] have done architecture optimization of a spatial 3-UPU parallel manipulator for maximizing the global conditioning index.

1.4 Motivation and Scope of the Thesis

The literature survey indicates that there exist significant gaps in the study of parallel and hybrid manipulators. Though some examples of application of different control strategies are available, there has not been any systematic study on comparison of different control strategies through various regulation and tracking problems in case of parallel manipulators. Secondly, unlike serial manipulators, where a lot of work has been done on path-planning, in case of parallel manipulators, research has been relatively scarce.

In this thesis, in the first part, we focus on implementation of different control strategies for controlling parallel and hybrid manipulators, and assessing their efficacy through various examples. We use a generalized Newton-Euler strategy for deriving the closed-form dynamic equations of these manipulators. In this thesis, we have considered PD control, model based control and optimal control schemes. In the second part of this thesis, we focus on path-planning of parallel manipulators. At first we discuss the problem of path-planning of parallel manipulators and then we use the Variational Approach for planning singularity-free paths for various manipulators.

This study of dynamics, control, and path planning is illustrated by several examples – we start with the simple two-degree-of-freedom 5-Bar planar manipulators with rotary (R) and prismatic (P) joints; we then discuss the more complicated three-degree-of-freedom planar parallel manipulators (3-RPR and 3-RRR); and finally we discuss the 3-PRPS hybrid manipulator. The main contributions of this thesis are as follows:

1. Implementation and comparison of three control schemes for control of parallel manipulators.

2. Implementation of Variational Approach for singularity-free path-planning of these manipulators.

1.5 Thesis Organization

In Chapter 2, we implement a general procedure, based on Newton-Euler formulation, for deriving the equations of motion for all the parallel and hybrid manipulators studied in this work. Then,

we discuss the features of the three control strategies studied in this work, and based on a number of numerical simulations make a qualitative comparison between them. In Chapter 3, we present the use of force transfromation matrix to determine the singularities of parallel and hybrid manipulators. Next, we formulate the problem of path planning for parallel manipulators and present the variational approach for path planning of these manipulators. Finally in Chapter 4, we summarize our work and draw the relevant conclusions.

Chapter 2

Control of Parallel Manipulators

2.1 Introduction

From the Literature Survey in Chapter 1, we observe that research in the area of control has been somewhat limited, and there has not been any comprehensive study on comparison of the commonly used control strategies. The dynamics of parallel manipulators is highly nonlinear due to the coupling of the legs. So efficient control of parallel manipulators warrants design of robust control algorithms.

Among the many controllers, independent joint controllers (PD type) are usually used in industrial robot manipulators, because they are computationally very efficient and are easy to implement. However, independent joint controllers cannot achieve a satisfactory performance due to their inherent low rejection of disturbances and parameter variations. Because of such limitations, model-based control algorithms were proposed that have the potential to perform better than independent joint controllers (which do not account for manipulator dynamics). Another advantage of model based control is that parametric uncertainty can be somewhat modelled in the control law. Another class of PD control is coupled PD control, where a linearised model of the nonlinear system of equations is controlled, and control gains can be so chosen so as to fix the closed loop poles at some point. Optimal control is another strategy in which the linearised model (as in coupled PD control) is controlled by choosing the gains so as to minimise some performance measure (typically chosen as a weighted sum of the error and expenditure of energy used for the control action).

For implementing each of the above mentioned control strategies effectively, we need to derive the closed-form dynamic equations. While Lagrange-Euler formulation is better suited for deriving the closed-form equations than the Newton-Euler formulation in case of serial manipulators, past research by Driels et al [28] has shown that in case of parallel manipulators, Newton-Euler formulation can be used more effectively for dynamic formulation. Dasgupta [29] has also shown that in case of parallel manipulators, task-space formulation is more useful than joint-space formulation. In view of these two aspects, the Newton-Euler formulation has been successfully applied for quite a number of parallel manipulators by Giordano and Benea [30], and, Dasgupta

and Mruthyunjaya [31]. In this thesis, we use a generalized Newton-Euler strategy developed by Dasgupta and Choudhury [9] for deriving the closed form dynamic equations.

In this chapter, we discuss three control strategies : simple PD Control, Model-based Control and Optimal Control for controlling some parallel and hybrid manipulators for regulation and tracking purposes. We outline the basic features of each control strategy and for each one of them, we show an example where they are implemented. Finally we make a comparison between these three control strategies through an example.

In section 2.2, we use the Newton-Euler strategy for deriving the dynamic equations of various manipulators. In section 2.3, the three control strategies are discussed. In section 2.4, some simulations are shown where the control strategies are applied separately. We also consider a case where all the three strategies are applied, and a qualitative comparison is made based on their performances.

2.2 Closed Form Dynamic Equations

2.2.1 Algorithm for Dynamic Formulation

The algorithmic steps for deriving the closed-form dynamic equations of parallel manipulators, using the generalized Newton-Euler formulation are as follows:

1. For each leg,

   (a) Find the position, velocity and acceleration of the platform-connection-point in terms of the task-space coordinates and their derivatives.

   (b) Solve the kinematics of the leg for position, velocity and acceleration.

   (c) Let m be the number of links in series in the leg. Let l_1, l_2, $\cdots$, l_m denote the m links and j_1, j_2, $\cdots$, j_m denote the joints, starting from the base. Then, for $i = 1$ to m, consider the equilibrium of links l_{m-i+1} to l_m and take the component(s) of the force(s) or moment(s) corresponding to the freedom(s) allowed by joint j_{m-i+1}.

   (d) From the equations obtained above, express the reaction in terms of the actuations and the platform accelerations $\ddot{\mathbf{X}}$ for deriving the closed-form dynamic equations.

2. Consider equilibrium of the platform and write Newton's and Euler's equations for it. Simplify the equations to the standard form resulting in the closed-form dynamic equations.

2.2.2 Planar Manipulators

In this section, the dynamic equations are presented for four planar manipulators, namely 5-bar 2-dof manipulators with prismatic and revolute actuations, and 8-bar 3-dof manipulators with prismatic and revolute actuations.

5-Bar Planar Parallel Manipulator with Prismatic Actuations

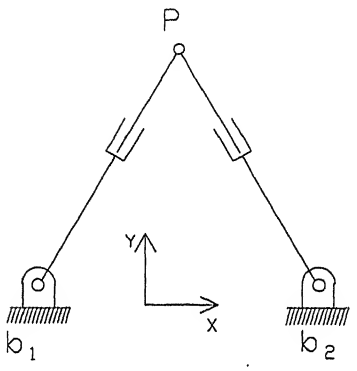


Figure 2.1: 5-Bar 2-dof Manipulator with Prismatic Actuations.

The 5-bar mechanism shown in figure 2.1 is a 2-dof manipulator with the end-effector at point 'p', the position of which is denoted by $\mathbf{t}$ and gives the generalized coordinates. In this case, the output-point itself is the platform-connection-point (for the point-platform). The closed form dynamic equations are compactly written as:

$$\mathbf{M}\ddot{\mathbf{t}} + \boldsymbol{\eta} = \mathbf{H}\mathbf{f} + \mathbf{R}_{Ext} \tag{2.1}$$

where $\mathbf{M} = \sum_{i=1}^{2} \mathbf{Q}_i$; $\boldsymbol{\eta} = -\sum_{i=1}^{2} \mathbf{U}_i$; $\mathbf{H} = [\ \mathbf{s}_1 \quad \mathbf{s}_2\ ]$ and $\mathbf{f} = [\ f_1 \quad f_2\ ]^T$.

$\mathbf{Q}$ and $\mathbf{U}$ are given by:

$$\mathbf{Q} = m_u \mathbf{s}\mathbf{s}^T + \frac{a_1}{L^2}\mathbf{s}_\perp \mathbf{s}_\perp{}^T + \frac{m_u(\mathbf{r}_u \times \mathbf{s})}{L}(\mathbf{s}\mathbf{s}_\perp{}^T + \mathbf{s}_\perp \mathbf{s}^T)$$

and

$$\begin{aligned}\mathbf{U} \;=\; & \left[m_u \mathbf{s}\cdot\mathbf{g} + m_u W^2 \mathbf{s}\cdot\mathbf{r}_u - m_u W^2 L + \frac{2m_u W\dot{L}(\mathbf{r}_u \times \mathbf{s})}{L} \right]\mathbf{s} \\ & + \frac{1}{L}\left[m_u \mathbf{r}_u \times \mathbf{g} + m_d \mathbf{r}_d \times \mathbf{g} + \frac{2a_1 W\dot{L}}{L} - m_u W^2 L(\mathbf{r}_u \times \mathbf{s}) + 2m_u W\dot{L}\mathbf{s}\cdot\mathbf{r}_u \right]\mathbf{s}_\perp\end{aligned}$$

in which

$$a_1 = m_u \mathbf{r}_u^2 + m_d \mathbf{r}_d^2 + I_u + I_d$$

The subscript $\perp$ denotes an anticlockwise rotation of a vector through a right angle. m_u and m_d are the masses of the lower and upper parts of each leg. $\mathbf{r}_u$ and $\mathbf{r}_d$ are the position vectors of centres of gravity of the lower and upper parts of each leg transformed to a fixed frame at the base point parallel to the global frame. I_u and I_d are the moments of inertia of the lower and upper part of each leg. W is the angular velocity of the entire leg and $\dot{L}$ is the sliding velocity of the prismatic joint. $\mathbf{R}_{Ext}$ is the external force acting at the end-effector. f_1 and f_2 are the forces exerted by the two actuators.

5-Bar Planar Parallel Manipulator with Revolute Actuations

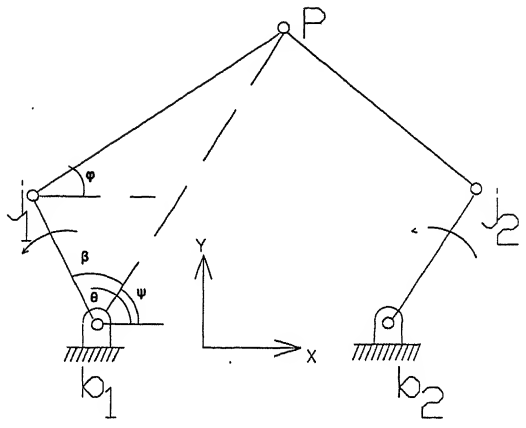


Figure 2.2: 5-Bar 2-dof Manipulator with Revolute Actuations.

Figure 2.2 shows a 2-dof planar parallel manipulator with revolute actuations. For the 2-dof parallel manipulator with revolute actuations, the closed form dynamic equations are given by equation (2.1), as in case of the 2-dof parallel manipulator with prismatic actuations.

The expressions for $\mathbf{M}$, $\boldsymbol{\eta}$, $\mathbf{H}$, and $\mathbf{f}$ are given by:

$$\mathbf{M} = \sum_{i=1}^{2} \mathbf{Q}_i; \quad \boldsymbol{\eta} = -\sum_{i=1}^{2} \mathbf{U}_i; \quad \mathbf{H} = \begin{bmatrix} \frac{\mathbf{d}_1}{l_1 \sin(\phi_1 - \theta_1)} & \frac{\mathbf{d}_2}{l_2 \sin(\phi_2 - \theta_2)} \end{bmatrix} \quad \text{and} \quad \mathbf{f} = \begin{bmatrix} \tau_1 & \tau_2 \end{bmatrix}^T$$

$\mathbf{Q}$ and $\mathbf{U}$ are given by:

$$\begin{aligned}
\mathbf{Q} &= \frac{1}{lu\sin^2(\phi-\theta)}[m_u l(\mathbf{d}\mathbf{d}^T u - \mathbf{c}\mathbf{r}_{u\perp}{}^T)\mathbf{c}_\perp \mathbf{d}^T - \mathbf{d}\mathbf{c}_\perp{}^T \mathbf{r}_{u\perp}\mathbf{c}^T \\
&\qquad + \frac{l}{u}(m_u \mathbf{r}_u^2 + I_u)\mathbf{c}\mathbf{c}^T + \frac{u}{l}(m_d \mathbf{r}_d^2 + I_d)\mathbf{d}\mathbf{d}^T] \\
\mathbf{G} &= m_u l(u\mathbf{d}\mathbf{c}_\perp{}^T - \mathbf{c}\mathbf{r}_{u\perp}{}^T) + m_d u \mathbf{d}\mathbf{r}_{d\perp}{}^T \\
\mathbf{V} &= m_u l(u\mathbf{d}\mathbf{c}_\perp{}^T - \mathbf{c}\mathbf{r}_{u\perp}{}^T)\mathbf{V}_u + m_d u \mathbf{d}\mathbf{r}_{d\perp}{}^T \mathbf{V}_d - \frac{I_d u}{l\sin(\phi-\theta)}\mathbf{d}\mathbf{d}^T\mathbf{V}_0 - \frac{I_u l}{u\sin(\phi-\theta)}\mathbf{c}\mathbf{c}^T\mathbf{V}_0 \\
\mathbf{U} &= \frac{\mathbf{G}\mathbf{g} - \mathbf{V}}{lu\sin(\phi-\theta)}
\end{aligned}$$

The quantities $\mathbf{V}_0$, $\mathbf{V}_d$ and $\mathbf{V}_u$ are as follows:

$$\mathbf{V}_0 = l\dot{\theta}^2\mathbf{c} + u\dot{\phi}^2\mathbf{d}$$

$$\mathbf{V}_d = \frac{\mathbf{r}_{d\perp}\mathbf{d}^T\mathbf{V}_o}{l\sin(\phi-\theta)} - \mathbf{r}_d\dot{\theta}^2$$

$$\mathbf{V}_u = \frac{\mathbf{c}_\perp\mathbf{d}^T\mathbf{V}_0}{\sin(\phi-\theta)} - \dot{\theta}^2 l\mathbf{c} - \mathbf{r}_u\dot{\phi}^2 - \frac{\mathbf{r}_{u\perp}\mathbf{c}^T\mathbf{V}_o}{u\sin(\phi-\theta)}$$

l and u are the lengths of the lower and upper links. m_d and m_u are their corresponding masses. I_u and I_d are the moments of inertia of the lower and upper links, respectively. $\mathbf{c}$ and $\mathbf{d}$ are the unit vectors along the lower and upper links. $\mathbf{r}_d$ and $\mathbf{r}_u$ denote the vector from the base point to the cg of the lower link and that from the intermediate joint to the cg of the upper link respectively. τ_1 and τ_2 are the torques acting in the lower legs of the two chains.

2.2.3 8-Bar Planar Parallel Manipulator with Prismatic Actuations

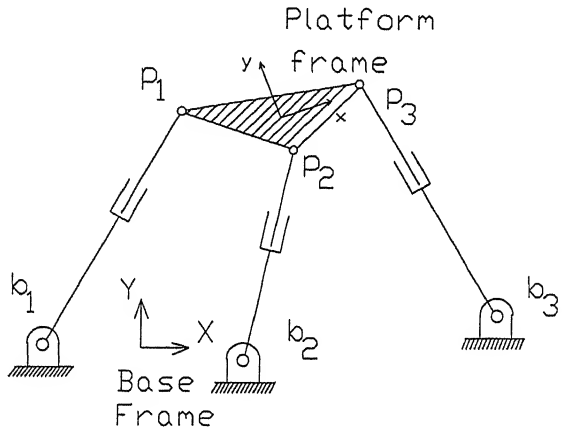


Figure 2.3: 8-Bar 3-dof Manipulator with Prismatic Actuations.

Figure 2.3 shows an 8-Bar 3-dof planar parallel manipulator with prismatic actuations. It has one prismatic joint in each leg, and the end-points of the legs have revolute joints. The end-effector in this case is a platform which is connected to the legs at three points $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$. The closed form dynamic equations are given by:

$$\mathbf{M}\begin{bmatrix} \ddot{\mathbf{t}} \\ \alpha \end{bmatrix} + \boldsymbol{\eta} = \mathbf{H}\mathbf{f} + \mathbf{R}_{Ext} \tag{2.2}$$

where the block elements of the Mass matrix M are:

$$\mathbf{M}_{11} = M_p\mathbf{E}_2 + \sum_{i=1}^{3}\mathbf{Q}_i$$

$$\mathbf{M}_{12} = \mathbf{M}_{21}^T = -(M_p\mathbf{R}_\perp + \sum_{i=1}^{3}\mathbf{Q}_i\mathbf{q}_{i\perp})$$

$$\mathbf{M}_{22} = I_p + M_pR^2 + \sum_{i=1}^{3}\mathbf{q}_{i\perp}^T\mathbf{Q}_i\mathbf{q}_{i\perp}$$

and

$$\eta = \begin{bmatrix} M_p(-\omega^2\mathbf{R} - \mathbf{g}) - \sum_{i=1}^{3} \mathbf{U}_i' \\ -M_p\mathbf{R} \times \mathbf{g} - \sum_{i=1}^{3} \mathbf{q}_i \times \mathbf{U}_i' \end{bmatrix}$$

$$H = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 \\ \mathbf{q}_1 \times \mathbf{s}_1 & \mathbf{q}_2 \times \mathbf{s}_2 & \mathbf{q}_3 \times \mathbf{s}_3 \end{bmatrix}$$

$$\mathbf{f} = \begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix}^T$$

The expression for $\mathbf{U}'$ is given by:

$$\mathbf{U}' = \mathbf{U} + \mathbf{Q}\omega^2\mathbf{q}$$

Here, the expressions for $\mathbf{Q}$ and $\mathbf{U}$ are same as in the 5-Bar Manipulator with prismatic actuations. $\mathbf{R}$ denotes the position vector of the centre of gravity of the platform in a reference frame about a base point and parallel to the global frame, and R is the distance of the centre of gravity of the platform from the origin of the platform reference frame. ω is the angular velocity of the platform, and $\mathbf{q}$ denotes the coordinates of platform connection point in global frame. M_p denotes the mass of the platform, I_p denotes the centroidal moment of inertia of the platform and $\mathbf{E}_2$ is a 2×2 identity matrix.

2.2.4 8-Bar Planar Parallel Manipulator with Revolute Actuations

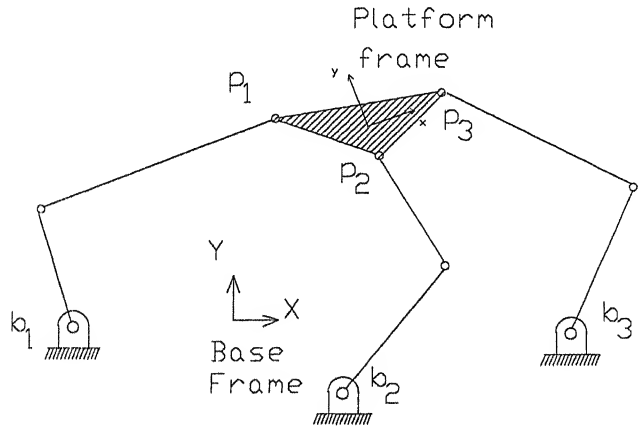


Figure 2.4: 8-Bar 3-dof Manipulator with Revolute Actuations.

Figure 2.4 shows an 8-Bar 3-dof planar parallel manipulator with revolute actuations. It has three revolute joints in each leg. The closed form dynamic equations, the matrix $\mathbf{M}$, and the vectors η and $\mathbf{f}$ are same as those in the 8-bar manipulator with prismatic actuations. Here, the

expressions for $\mathbf{Q}$ and $\mathbf{U}$ are same as those in the 5-Bar Manipulator with revolute actuations. The force transformation matrix $\mathbf{H}$ is given by:

$$\mathbf{H} = \begin{bmatrix} \frac{\mathbf{d}_1}{l_1 \sin(\phi_1 - \theta_1)} & \frac{\mathbf{d}_2}{l_2 \sin(\phi_2 - \theta_2)} & \frac{\mathbf{d}_3}{l_3 \sin(\phi_3 - \theta_3)} \\ \frac{\mathbf{q}_1 \times \mathbf{d}_1}{l_1 \sin(\phi_1 - \theta_1)} & \frac{\mathbf{q}_2 \times \mathbf{d}_2}{l_2 \sin(\phi_2 - \theta_2)} & \frac{\mathbf{q}_3 \times \mathbf{d}_3}{l_3 \sin(\phi_3 - \theta_3)} \end{bmatrix}$$

2.2.5 Spatial Manipulators

In this section, the Newton-Euler strategy for dynamic equations is applied to a 3-PRPS Hybrid Manipulator which is a 6-dof spatial manipulator.

3-PRPS Hybrid Manipulator

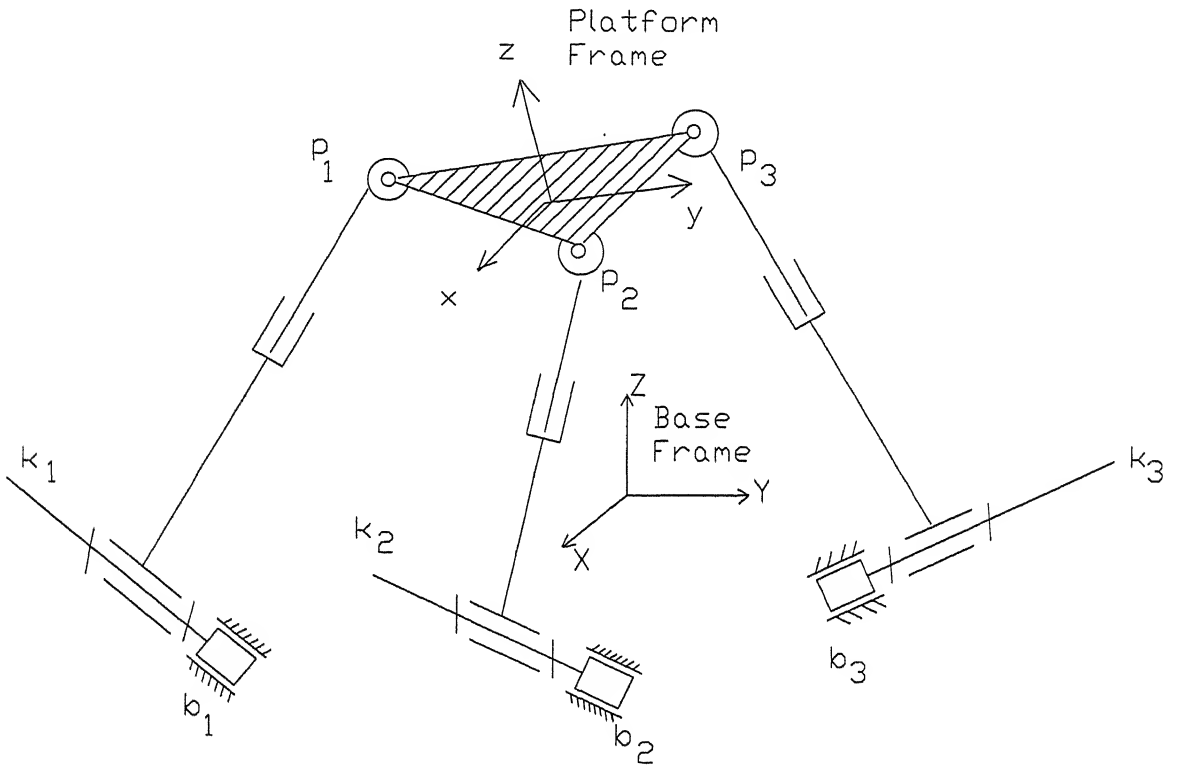


Figure 2.5: 3-PRPS Hybrid Manipulator.

Figure 2.5 shows a 3-PRPS Hybrid Manipulator. This manipulator has two prismatic actuations in each leg. The equations of motion are given by:

$$\mathbf{M} \begin{bmatrix} \ddot{\mathbf{t}} \\ \boldsymbol{\alpha} \end{bmatrix} + \boldsymbol{\eta} = \mathbf{H}\mathbf{F} + \begin{bmatrix} \mathbf{R}_{Ext} \\ \mathbf{M}_{Ext} \end{bmatrix} \tag{2.3}$$

where

$$\mathbf{M} = \begin{bmatrix} M_p \mathbf{E}_3 + \sum_{i=1}^{3} \mathbf{Q}_i & -(M_p \tilde{\mathbf{R}} + \sum_{i=1}^{3} \mathbf{Q}_i \tilde{\mathbf{q}}_i) \\ M_p \tilde{\mathbf{R}} + \sum_{i=1}^{3} \tilde{\mathbf{q}}_i \mathbf{Q}_i & \mathbf{I}_p + M_p(\mathbf{R}^2 \mathbf{E}_3 - \mathbf{R}\mathbf{R}^T) + \sum_{i=1}^{3} \tilde{\mathbf{q}}_i \mathbf{Q}_i \tilde{\mathbf{q}}_i \end{bmatrix}$$

$$
\begin{aligned}
\eta &= \begin{bmatrix} M_p\{\omega \times (\omega \times \mathbf{R}) - \mathbf{g}\} - \sum_{i=1}^{3} \mathbf{U}_i \\ \omega \times (\mathbf{I}_p \omega) + M_p \mathbf{R} \times \{(\omega \cdot \mathbf{R})\omega - \mathbf{g}\} - \sum_{i=1}^{3} \mathbf{q}_i \times \mathbf{U}_i \end{bmatrix} \\
\mathbf{H} &= \begin{bmatrix} \mathbf{k}_1 & \mathbf{k}_2 & \mathbf{k}_3 & \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 \\ \mathbf{q}_1 \times \mathbf{k}_1 & \mathbf{q}_2 \times \mathbf{k}_2 & \mathbf{q}_3 \times \mathbf{k}_3 & \mathbf{q}_1 \times \mathbf{s}_1 & \mathbf{q}_2 \times \mathbf{s}_2 & \mathbf{q}_3 \times \mathbf{s}_3 \end{bmatrix} \\
\mathbf{F} &= \begin{bmatrix} f_{k_1} & f_{k_2} & f_{k_3} & f_{s_1} & f_{s_2} & f_{s_3} \end{bmatrix}^T
\end{aligned}
$$

The expressions for $\mathbf{Q}$ and $\mathbf{U}$ are given by:

$$
\begin{aligned}
\mathbf{Q} &= m_u \mathbf{s}\mathbf{s}^T + \frac{m_u}{L}\{\mathbf{s} \cdot (\mathbf{k} \times \mathbf{r}_u)\}\{\mathbf{s}\mathbf{n}^T + \mathbf{n}\mathbf{s}^T\} + (m_u + m_m + m_d)\mathbf{k}\mathbf{k}^T + \frac{a_1}{L^2}\mathbf{n}\mathbf{n}^T \\
\mathbf{U} &= [m_u(\mathbf{s} \cdot \mathbf{g}) - m_u \mathbf{s} \cdot \mathbf{W} \times (\mathbf{W} \times \mathbf{r}_u) - m_u u_3 - \frac{m_u}{L}\{\mathbf{s} \cdot (\mathbf{k} \times \mathbf{r}_u)\}u_4]\mathbf{s} \\
&\quad + [(m_u + m_m + m_d)(\mathbf{k} \cdot \mathbf{g}) - \{m_u \mathbf{W} \times (\mathbf{W} \times \mathbf{r}_u) + m_m \mathbf{W} \times (\mathbf{W} \times \mathbf{r}_m) \\
&\quad + m_d \mathbf{W} \times (\mathbf{W} \times \mathbf{r}_d)\} \cdot \mathbf{k} - 2m_u \dot{L}\mathbf{k} \cdot (\mathbf{W} \times \mathbf{s}) - (m_u + m_m)u_1 - m_u(\mathbf{k} \cdot \mathbf{s})u_3]\mathbf{k} \\
&\quad + \frac{1}{L}[m_u \mathbf{k} \cdot (\mathbf{r}_u \times \mathbf{g}) + m_m \mathbf{k} \cdot (\mathbf{r}_m \times \mathbf{g}) - \mathbf{k} \cdot \{\mathbf{W} \times (\mathbf{I}_u + \mathbf{I}_m)\mathbf{W}\} - m_u \mathbf{k} \cdot (\mathbf{r}_u \times \mathbf{s})u_3 - a_1 u_4]
\end{aligned}
$$

where

$$
a_1 = m_m(\mathbf{k} \times \mathbf{r}_m)^2 + m_u(\mathbf{k} \times \mathbf{r}_u)^2 + \mathbf{k} \cdot (\mathbf{I}_u + \mathbf{I}_m)\mathbf{k}
$$

and

$$
\mathbf{n} = \mathbf{k} \times \mathbf{s}
$$

In the above expressions, the notation $\tilde{\mathbf{q}}$ refers to the skew-symmetric matrix form of $\mathbf{q}$, and the notation is same for all other vectors. $\mathbf{r}_m$ denotes the position of the centre of gravity of the middle leg (in global frame), and m_m and $\mathbf{I}_m$ are the mass and inertia matrix (in global frame) of the middle leg, respectively. The directions of the prismatic joints at the base are given by the vector $\mathbf{k}$. All other notations are similar to the ones discussed earlier.

2.3 Control Strategies

Most of the present day control strategies use feedback to compute servo error by finding the difference between the desired position and actual position, and likewise between desired and actual velocity. In this section, we discuss three such feedback control strategies, namely, PD control, Model-based control and Optimal Control which we will be employing for controlling the parallel manipulators.

2.3.1 PD Control

In **PD control**, a linear approximation of the nonlinear model is made and the manipulator is controlled based on this linear model. Manipulator control problem is a **multi-input, multi-output (MIMO)** control problem. In PD control, one can implement either **coupled control**, where one designs a single **MIMO** control system, or one can implement **decoupled control**, where one designs N **single-input, single-output (SISO)** control systems.

Decoupled PD Control

In a decoupled PD control scheme, we construct a control system by treating each joint as a separate system to be controlled (in Joint-space control), or by controlling each component of position and/or orientation separately (in Task-space control).

For parallel manipulators, for Joint-space control, the control law for the i'th actuator is given by:

$$F_i = Kp_i(L_{i_0} - L_i) - Kv_i\dot{L}_i \tag{2.4}$$

where, L_{i_0} is the desired actuator length[1] and L_i and $\dot{L}_i$ are the actual leg-length and leg-velocity of the i'th actuator, and Kp_i and Kv_i are the control gains. For Task-space control, the decoupled PD control law for controlling the task-space variable t, is given by:

$$F = Kp(t_0 - t) - Kv\dot{t} \tag{2.5}$$

By choosing our control gains appropriately, we can make the closed loop system to be as any second order system that we wish. Often, gains are chosen so as to obtain critical damping or slight over damping.

Coupled PD Control

In closed form, the manipulator dynamics is given by a nonlinear time-dependent equation:

$$\mathbf{M}\ddot{\mathbf{v}} + \boldsymbol{\eta} = \mathbf{Hf} + \mathbf{R}_{Ext} \tag{2.6}$$

where, $\mathbf{M}$ is the mass matrix, $\boldsymbol{\eta}$ is the vector consisting of centrifugal and coriolis forces, $\mathbf{H}$ is the force transformation matrix, $\mathbf{f}$ is the vector of joint forces/torques and $\mathbf{R}_{Ext}$ is the vector of external forces and/or moments acting at the end-effector. In case of 5-bar 2-dof manipulators, $\mathbf{v} = \mathbf{t}$; in case of 8-bar 3-dof manipulators, $\mathbf{v} = [\mathbf{t} \quad \theta]^T$; and in case of 3-PRPS hybrid manipulator, $\mathbf{v} = [\mathbf{t} \quad \boldsymbol{\theta}]^T$.

Defining the state vector $\mathbf{z} = [\mathbf{v} \quad \dot{\mathbf{v}}]^T$, equation (2.6) can be re-written in the form:

$$\dot{\mathbf{z}} = \begin{bmatrix} \dot{\mathbf{v}} \\ \mathbf{M}^{-1}[\mathbf{Hf} - \boldsymbol{\eta} + \mathbf{R}_{Ext}] \end{bmatrix} \tag{2.7}$$

The above equation can be written in the form:

$$\dot{\mathbf{z}} = \mathbf{g}(\mathbf{z}, \mathbf{F}) \tag{2.8}$$

where $\mathbf{F}$ stands for $\mathbf{f}$ in joint-space control, but for $\mathbf{Hf}$ in task-space control. At each instant, equation (2.8) can be linearised in the form:

$$\dot{\mathbf{z}} = \mathbf{Az} + \mathbf{BF}$$

[1] or other joint variable, as the case may be

where $\mathbf{z}$ = state-vector,

$\mathbf{F}$ = control-vector,

$\mathbf{A}$ = Jacobian-Matrix corresponding to the state-vector,

$\mathbf{B}$ = Jacobian-Matrix corresponding to the control-vector.

In coupled control, we choose the control-vector to be:

$$\mathbf{F} = -\mathbf{K}\mathbf{z} \tag{2.9}$$

The stability and transient response characteristics are determined by the eigen-values of the matrix $\mathbf{A} - \mathbf{BK}$. If the matrix $\mathbf{K}$ is chosen properly, then the matrix $\mathbf{A} - \mathbf{BK}$ can be made asymptotically stable matrix. The eigen-values of the matrix $\mathbf{A} - \mathbf{BK}$ are the desired closed loop poles. If these poles are in the left-half of the s-plane, then the control strategy is successful. From the Matrix $\mathbf{K}$, one can determine the position gain matrix $\mathbf{Kp}$ and the velocity gain matrix $\mathbf{Kv}$.

For Joint-space control, the control law is given by:

$$\mathbf{F} = \mathbf{Kp}(\mathbf{L}_0 - \mathbf{L}) - \mathbf{Kv}\dot{\mathbf{L}} \tag{2.10}$$

where, $\mathbf{L}_0$ is the vector of desired actuator lengths and $\mathbf{L}$ and $\dot{\mathbf{L}}$ are the actual vectors of leg-lengths and leg-velocities, respectively.

For Task-space control, the coupled PD control law for controlling the task-space vector $\mathbf{t}$, is given by:

$$\mathbf{F} = \mathbf{Kp}(\mathbf{t}_0 - \mathbf{t}) - \mathbf{Kv}\dot{\mathbf{t}} \tag{2.11}$$

where $\mathbf{t}_0$ is the desired task-space vector.

2.3.2 Model-Based Control

In Model-Based control, we employ control-law partitioning. In this method, we partition the controller into a **model-based portion** and a **servo portion**. The result is that the system-parameters appear only in the model-based portion, and the servo portion is independent of these parameters. The control law takes the form:

$$\mathbf{F} = \boldsymbol{\alpha}\mathbf{F}' + \boldsymbol{\beta} \tag{2.12}$$

where, for a system of n degrees of freedom, $\mathbf{F}$, $\mathbf{F}'$, and $\boldsymbol{\beta}$ are $n \times 1$ vectors, and $\boldsymbol{\alpha}$ is an $n \times n$ matrix. The matrix $\boldsymbol{\alpha}$ is chosen to **decouple** the n equations of motion. If $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are properly chosen, then from $\mathbf{F}'$ input, the system appears to be n independent unit masses. For this reason, in the multidimensional case, the model-based portion of the control law is called a **linearizing and decoupling** control law.

The servo law for trajectory following of a multidimensional system becomes:

$$\mathbf{F}' = \ddot{\mathbf{X}}_d + \mathbf{K}_v\dot{\mathbf{E}} + \mathbf{K}_p\mathbf{E}, \tag{2.13}$$

where $\mathbf{K}_v$ and $\mathbf{K}_p$ are now $n \times n$ matrices, which are generally chosen to be diagonal with constant gains on the diagonal. $\mathbf{E}$ and $\dot{\mathbf{E}}$ are $n \times 1$ vectors of errors in actuator lengths and actuator velocites (in case of Joint-space control) and vectors of errors in position and velocity of end-effector (in case of Task-space control). $\ddot{\mathbf{X}}_d$ is the vector of desired actuator accelerations (in case of Joint-space control) and desired acceleration of end-effector (in case of Task-space control).

The servo law for regulation of a multidimensional system becomes:

$$\mathbf{F}' = \mathbf{K}_v\dot{\mathbf{E}} + \mathbf{K}_p\mathbf{E}. \tag{2.14}$$

Choice of α and β for Parallel Manipulators

The closed form dynamic equations of a parallel manipulator are:

$$\mathbf{M}\ddot{\mathbf{X}} + \boldsymbol{\eta} = \mathbf{H}\mathbf{F} + \mathbf{R}_{Ext}$$

where, $\mathbf{M}$ is the Inertia Matrix, $\boldsymbol{\eta}$ is the vector of centrifugal and coriolis terms, $\mathbf{R}_{Ext}$ is the vector of external forces and/or moments, $\mathbf{H}$ is the force-transfomation matrix, and $\mathbf{F}$ is the vector of actuator/joint forces/torques. $\ddot{\mathbf{X}}$ is the vector of linear and angular accelerations. $\mathbf{F}$ can be re-written as:

$$\mathbf{F} = \mathbf{H}^{-1}\mathbf{M}\ddot{\mathbf{X}} + \mathbf{H}^{-1}(\boldsymbol{\eta} - \mathbf{R}_{Ext}) \tag{2.15}$$

From the above equation, one can choose $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as:

$$\begin{aligned} \boldsymbol{\alpha} &= \mathbf{H}^{-1}\mathbf{M} \\ \boldsymbol{\beta} &= \mathbf{H}^{-1}(\boldsymbol{\eta} - \mathbf{R}_{Ext}) \end{aligned}$$

Substituting the values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in equation (2.12) and using equation (2.15) we get:

$$\ddot{\mathbf{X}} = \mathbf{F}' \tag{2.16}$$

Lack of knowledge of parameters

In all the previous discussions, we have assumed that the parameter values are exact. But in practice, there is always some error in the knowledge of the parameters. Moreover, if the manipulator has some portion of its dynamics which are not repeatable, because, for example, they change as the robot ages, it is difficult to have good parameter values in the model at all times. If we are to incorporate this into our model-based controller, then equation (2.16) must be changed.

Let us assume that $\mathbf{M}$, $\mathbf{H}$ and $\boldsymbol{\eta}$ are calculated based on the *nominal* parameter values, and are used in the control law (equation (2.15)). Further, we denote the corresponding *actual* quantities by $\mathbf{M}_1$, $\mathbf{H}_1$ and $\boldsymbol{\eta}_1$, respectively, which are computed based on the values of dynamic parameters determined by random variation within $\pm$ 5% of the *nominal* values.

Then the new combined equation will be given by the equation

$$\ddot{\mathbf{X}} = (\mathbf{M}_1^{-1}\mathbf{H}_1\mathbf{H}^{-1}\mathbf{M})\mathbf{F}' + \mathbf{M}_1^{-1}\mathbf{H}_1\mathbf{H}^{-1}(\boldsymbol{\eta} - \mathbf{R}_{Ext}) + \mathbf{M}_1^{-1}(\mathbf{R}_{Ext} - \boldsymbol{\eta}_1) \tag{2.17}$$

where the expression for $\mathbf{F}'$ is the same as before. We use this equation for the simulation of the model-based control against parameter uncertainty.

2.3.3 Optimal Control

In Optimal Control, the closed-form nonlinear dynamics equation is linearised as in case of coupled PD control. Here, our intention is to determine the matrix $\mathbf{K}$ which will minimise the performance index J which is given by:

$$J = \int_0^\infty (\mathbf{z}^T\mathbf{Q}\mathbf{z} + \mathbf{F}^T\mathbf{R}\mathbf{F})dt \tag{2.18}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are real symmetric matrices. Note that the second term of this integral accounts for the expenditure of the energy of control signals. The matrices $\mathbf{Q}$ and $\mathbf{R}$ determine the relative importance of the error and the expenditure of this energy. The design of optimal control systems based on these types of quadratic performance indices boils down to the determination of the elements of the matrix $\mathbf{K}$. By setting

$$\mathbf{z}^T(\mathbf{Q} + \mathbf{K}^T\mathbf{R}\mathbf{K})\mathbf{z} = -\frac{d}{dt}(\mathbf{z}^T\mathbf{P}\mathbf{z}) \tag{2.19}$$

where $\mathbf{P}$ is a positive-definite Hermitian or real symmetric matrix, and by doing some mathematical manipulations it can be shown [32] that the optimal matrix $\mathbf{K}$ is given by:

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} \tag{2.20}$$

For determining this matrix, we need to solve the reduced-matrix Riccati equation for determining $\mathbf{P}$. This equation is given by:

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = \mathbf{0} \tag{2.21}$$

For applying Optimal Control in case of parallel manipulators, at each time-step, the optimal matrix $\mathbf{K}$ has to be calculated (by solving the Riccati equation), and the matrices $\mathbf{Kp}$ and $\mathbf{Kv}$ are derived from it. The control laws are same as those used for coupled PD control (equations (2.10) and (2.11)).

2.4 Results and Discussions

The dynamic formulation of the manipulators considered have been implemented in MATLAB routines for forward dynamics (simulations). The programs developed for forward dynamics

simulation use the MATLAB routine "ode45" (which is based on the 4th and 5th order Runge-Kutta formulae with adaptive step-size) for solving the system of differential equations. The dynamic formulations have been applied for all the control strategies discussed in this Chapter. Some simulation results are presented below as illustrations. Based on the simulation results, we make some conclusions about the efficacies of each of the control strategies discussed. For all the cases, external loads ($\mathbf{F}_{Ext}$ and $\mathbf{M}_{Ext}$) are taken as zero.

2.4.1 Example I

As the first example, let us consider a 5-Bar Manipulator with prismatic actuations. Through this example we make a **comparison between decoupled and coupled PD control**. The kinematic and dynamic parameters are given in Appendix I. Here, a regulation problem has been considered. The errors in task-space are used to determine the required force at the end-effector (a point in this case) and the necessary actuator forces are obtained by operating with $\mathbf{H}^{-1}$, which is equivalent to the Jacobian-transpose. Initial conditions taken for the simulation are

$$\mathbf{t}_0 = \begin{bmatrix} 0.34 & 0.5 \end{bmatrix}^T \text{m}$$

with zero velocity, while the purpose of the control is to regulate the end-point at the desired position

$$\mathbf{t}_d = \begin{bmatrix} 0.45 & 0.65 \end{bmatrix}^T \text{m}$$

The duration of simulation is $T = 1.0$ sec. For decoupled control, the control gains are $k_p = 1537.9$ and $k_v = 80$. For coupled control, the matrices $\mathbf{K}_p$ and $\mathbf{K}_v$ are evaluated at each time-step by using the MATLAB function "place(A,B,L)", where A and B are the Jacobian Matrices corresponding to the state-vector and the force-vector respectively, and $\mathbf{L}$ is the vector of desired closed-loop poles. The vector $\mathbf{L}$ is chosen as

$$\mathbf{L} = \begin{bmatrix} -20 & -30 & -40 & -100 \end{bmatrix}$$

The simulation results (along with the desired state) in task-space are shown in figures 2.6 and 2.7 respectively. It is seen that the regulation is successful in both cases, but in coupled control the regulation is achieved in a much shorter span than in decoupled control, where the response is oscillatory in nature. The performance of the decoupled control scheme can be improved upon by choosing better control gains by trial and error. On the other hand the gain matrices, for coupled control, are chosen exactly by using the function "place" and hence its performance is better. But computation is much more involved in the latter case, as the gain matrices must be evaluated at each time-step.

2.4.2 Example II

In the second example, we consider a 8-Bar 3-dof manipulator with revolute actuations. Through this example we make a **comparison between Task-space and Joint-space control**. The

kinematic and dynamic parameters are given in Appendix I. Here, a regulation problem has been considered. The errors in task-space are used to determine the required force at the end-effector (platform) in case of task-space control using the decoupled PD control law given in equation (2.5). In case of joint-space control, the errors in actuator-lengths are obtained by inverse-kinematics at each time-step and the decoupled PD control law given in equation (2.4) is used. Initial conditions taken for the simulation are

$$\mathbf{t}_0 = \begin{bmatrix} 0.3 & 0.1 \end{bmatrix}^T \text{m} \qquad \theta_0 = 0.3 \text{ rad}$$

with zero velocity, while the purpose of the control is to regulate the end-point at the desired position

$$\mathbf{t}_d = \begin{bmatrix} 0.5 & 0.25 \end{bmatrix}^T \text{m} \qquad \theta_d = 0.1 \text{ rad}$$

The duration of simulation is $T = 1.0$ sec. For decoupled control, the control gains are $k_p = 1537.9$ and $k_v = 80$. The simulation results (along with the desired state) in task-space are shown in figures 2.8 and 2.9 respectively. It is seen that the regulation is successful in both cases, but in task-space control the regulation is achieved in a much shorter span than in joint-space control. Also, since in parallel manipulators task-space variables are the natural choice for generalized coordinates, so in most cases task-space control is expected to give better results than joint-space control. Moreover, for a joint-space control scheme, it would be essential to select the control gains very carefully, as otherwise the integration process may run into serious numerical catastrophe leading to both loss of precision and change of branch in task-space. Computationally joint-space control is more involved as at each stage inverse kinematics is to be done. So task-space control is better and more stable than joint-space control.

2.4.3 Example III

In the third example, we consider a 3-PRPS spatial manipulator, details of which are given in Appendix I. Here, a regulation problem and a tracking problem have been considered with **Model-based control** in task-space. We have assumed parametric uncertainty (varying between -5% to +5%) in both the cases.

Initial conditions taken for simulation of the regulation problem are

$$\mathbf{t}_0 = \begin{bmatrix} 0.2 & 0.2 & 0.4 \end{bmatrix}^T \text{m} \qquad \boldsymbol{\theta}_0 = \begin{bmatrix} 0.0 & 0.0 & -0.2 \end{bmatrix}^T \text{rad}$$

with zero velocity, while the purpose of the control is to regulate the end-point at the desired position

$$\mathbf{t}_d = \begin{bmatrix} 0.3 & 0.35 & 0.45 \end{bmatrix}^T \text{m} \qquad \boldsymbol{\theta}_d = \begin{bmatrix} 0.1 & 0.2 & 0.1 \end{bmatrix}^T \text{rad}$$

Initial conditions taken for simulation of the tracking problem are

$$\mathbf{t}_0 = \begin{bmatrix} 0.6 & 0.4 & 0.3 \end{bmatrix}^T \text{m} \qquad \boldsymbol{\theta}_0 = \begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T \text{rad}$$

with zero velocity, while the purpose of the control is to track a path. The path to be tracked is linear having parabolic bends at two ends. The maximum linear velocity is $V = 1.0$ m/s and the maximum angular velocity is $\Omega = 2.0$ rad/s. The end-point of the path is

$$\mathbf{t}_d = \begin{bmatrix} 0.5 & 0.5 & 0.4 \end{bmatrix}^T \text{m} \qquad \boldsymbol{\theta}_d = \begin{bmatrix} 0.2 & 0.1 & 0.1 \end{bmatrix}^T \text{rad}$$

In both the cases, the duration of simulation is $T = 1.0$ sec and, for the servo part of the control law, the control gains are $k_p = 1451.2$ and $k_v = 100$. The simulation results (along with the desired state) in task-space are shown in figures 2.10 and 2.11 respectively. In either of these cases model-based control is found to be very effective.

2.4.4 Example IV

In the fourth example, we consider the same 5-Bar manipulator, considered in the first example. Here, a regulation problem and a tracking problem have been considered with **Optimal control** in task-space.

Initial conditions taken for the simulation of the regulation problem are

$$\mathbf{t}_0 = \begin{bmatrix} 0.34 & 0.5 \end{bmatrix}^T \text{m}$$

with zero velocity, while the purpose of the control is to regulate the end-point at the desired position

$$\mathbf{t}_d = \begin{bmatrix} 0.45 & 0.65 \end{bmatrix}^T \text{m}$$

The duration of simulation is $T = 1.0$ sec. For optimal control, the matrices $\mathbf{K}_p$ and $\mathbf{K}_v$ are evaluated at each time-step by using the MATLAB function "lqr(A,B,Q,R)", where $\mathbf{A}$ and $\mathbf{B}$ are the Jacobian Matrices corresponding to the state-vector and the force-vector respectively, and $\mathbf{Q}$ and $\mathbf{R}$ are the matrices which determine the relative importance of the error and the expenditure of the energy of control signals. The matrices $\mathbf{Q}$ and $\mathbf{R}$ are given in Appendix II.

Initial conditions taken for the simulation of the tracking problem are

$$\mathbf{t}_0 = \begin{bmatrix} 0.3 & 0.3 \end{bmatrix}^T \text{m}$$

with zero velocity, while the purpose of the control is to track a path. The path to be tracked is linear having parabolic bends at two ends. The maximum linear velocity is $V = 1.0$ m/s and the maximum angular velocity is $\Omega = 1.0$ rad/s. The end-point of the path is

$$\mathbf{t}_d = \begin{bmatrix} 0.45 & 0.4 \end{bmatrix}^T \text{m}$$

The duration of simulation is $T = 1.0$ sec. The matrices $\mathbf{Q}$ and $\mathbf{R}$ are given in Appendix II.

The simulation results (along with the desired state) in task-space are shown in figures 2.12 and 2.13 respectively. From the results we see that optimal control works fine for regulation problem, but the results in the tracking problem are not so encouraging. The performance surely depends on the choice of the matrices $\mathbf{Q}$ and $\mathbf{R}$.

2.4.5 Example V

In this example, we consider a 8-Bar 3-dof manipulator with prismatic actuations. Through this example we make a **comparison between PD control, Model-based control and Optimal control.** The kinematic and dynamic parameters are given in Appendix I. Here, a tracking problem has been considered. In model-based control, we have assumed parametric uncertainty (varying between -5% to +5%).

Initial conditions taken for the simulation are

$$\mathbf{t}_0 = \begin{bmatrix} 0.3 & 0.3 \end{bmatrix}^T \text{m} \qquad \theta_0 = 0.1 \text{ rad}$$

with zero velocity, while the purpose of the control is to track a path. The path to be tracked is linear having parabolic bends at two ends. The maximum linear velocity is $V = 1.0$ m/s and the maximum angular velocity is $\Omega = 2.0$ rad/s. The end-point of tracking is given by

$$\mathbf{t}_d = \begin{bmatrix} 0.45 & 0.45 \end{bmatrix}^T \text{m} \qquad \theta_d = 0.5 \text{ rad}$$

The duration of simulation is $T = 1.0$ sec.

For decoupled PD control and model-based control, the control gains are $k_p = 1451.2$ and $k_v = 80$. The matrices $\mathbf{Q}$ and $\mathbf{R}$ are given in Appendix II. The simulation results (along with the desired state) in task-space are shown in figures 2.14, 2.15 and 2.16 respectively. From the results we see that performance of model-based control is the best amongst the three control strategies, in spite of parametric uncertainty considered only in that case.

2.4.6 Discussion

From the simulations of the different control strategies, we can conclude that model-based control performs best among the control strategies considered in the work. Model-based control is, however, computationally more expensive than decoupled PD control, as the dynamic quantities $\mathbf{M}$, $\mathbf{H}$ and $\boldsymbol{\eta}$, are to be computed at each time-step. Both coupled PD control and optimal control are also computationally expensive, as the gain matrices are to be derived at each time step. Comparatively, the computational burden of computing the dynamics is much less than that of the gain matrices at each time-step. One of the drawbacks of optimal control is that the matrices $\mathbf{Q}$ and $\mathbf{R}$ are chosen by trial and error — so the performance of the strategy is dependent on the choice of these matrices. If the control gains are chosen properly then decoupled PD control can be expected to perform commendably over short range of operation. Simulation in joint-space would involve heavy computational burden, which may lead to loss of precision. In addition, PD control law in joint-space may give rise to additional problem of branching of forward kinematics, unless the control gains are chosen very carefully and the manipulator parameters are known very accurately. Knowledge of kinematic parameters must be accurate in PD control and optimal control for achieving the desired response; but model based control works well even under parametric uncertainty.

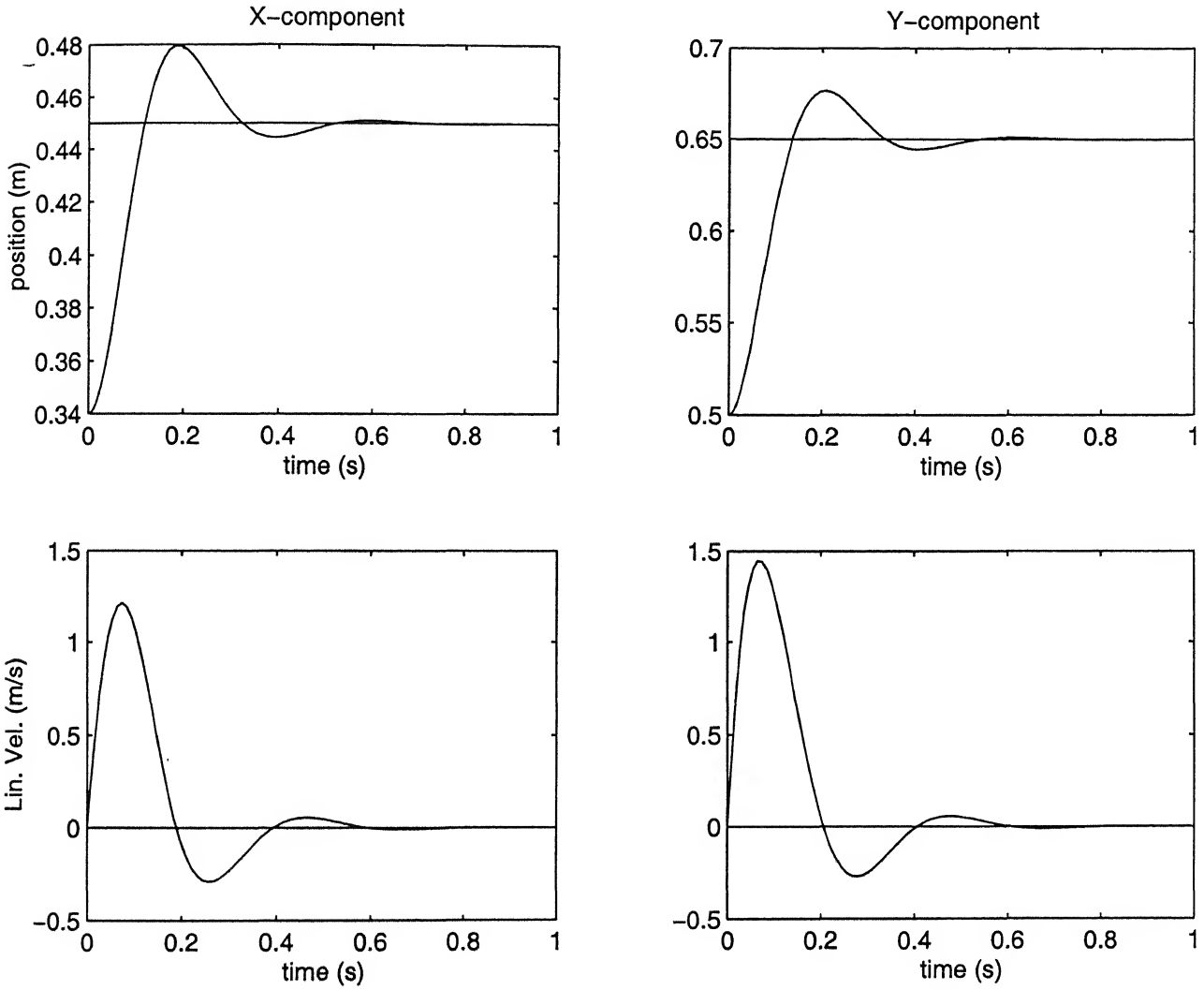


Figure 2.6: Decoupled PD Control of a 5-Bar Manipulator.

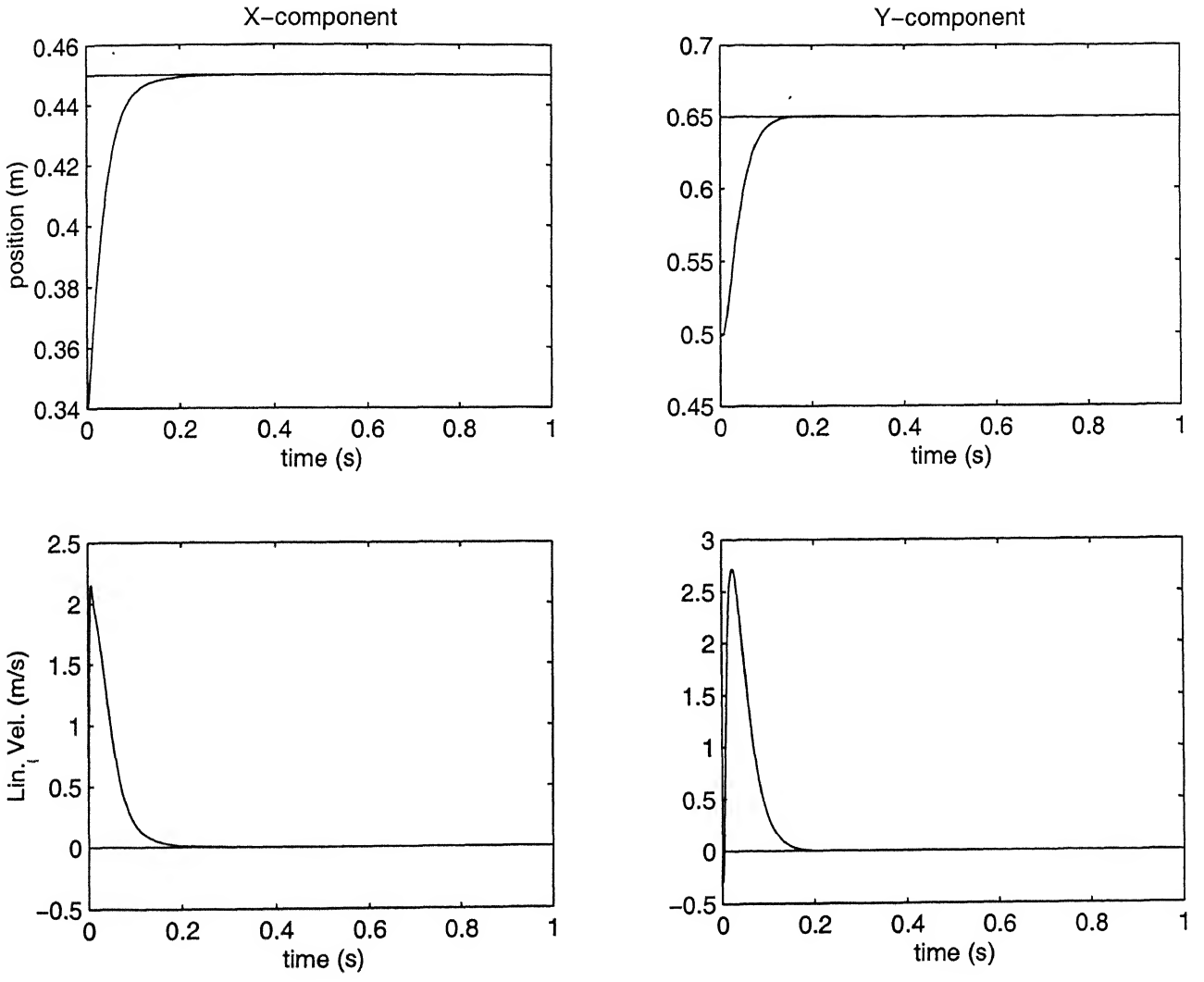


Figure 2.7: Coupled PD Control of a 5-Bar Manipulator.

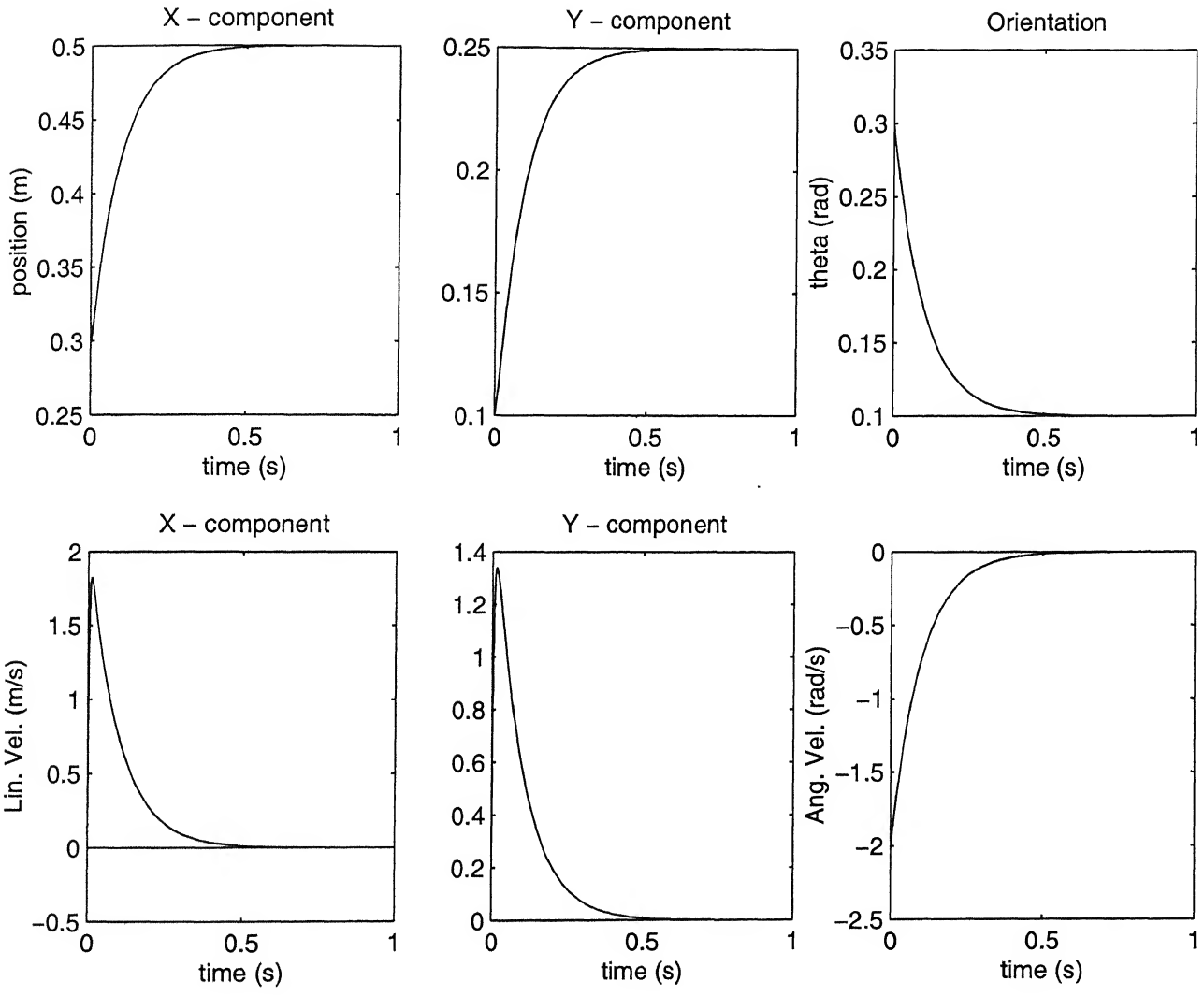


Figure 2.8: Task-space control of a 8-Bar Manipulator.

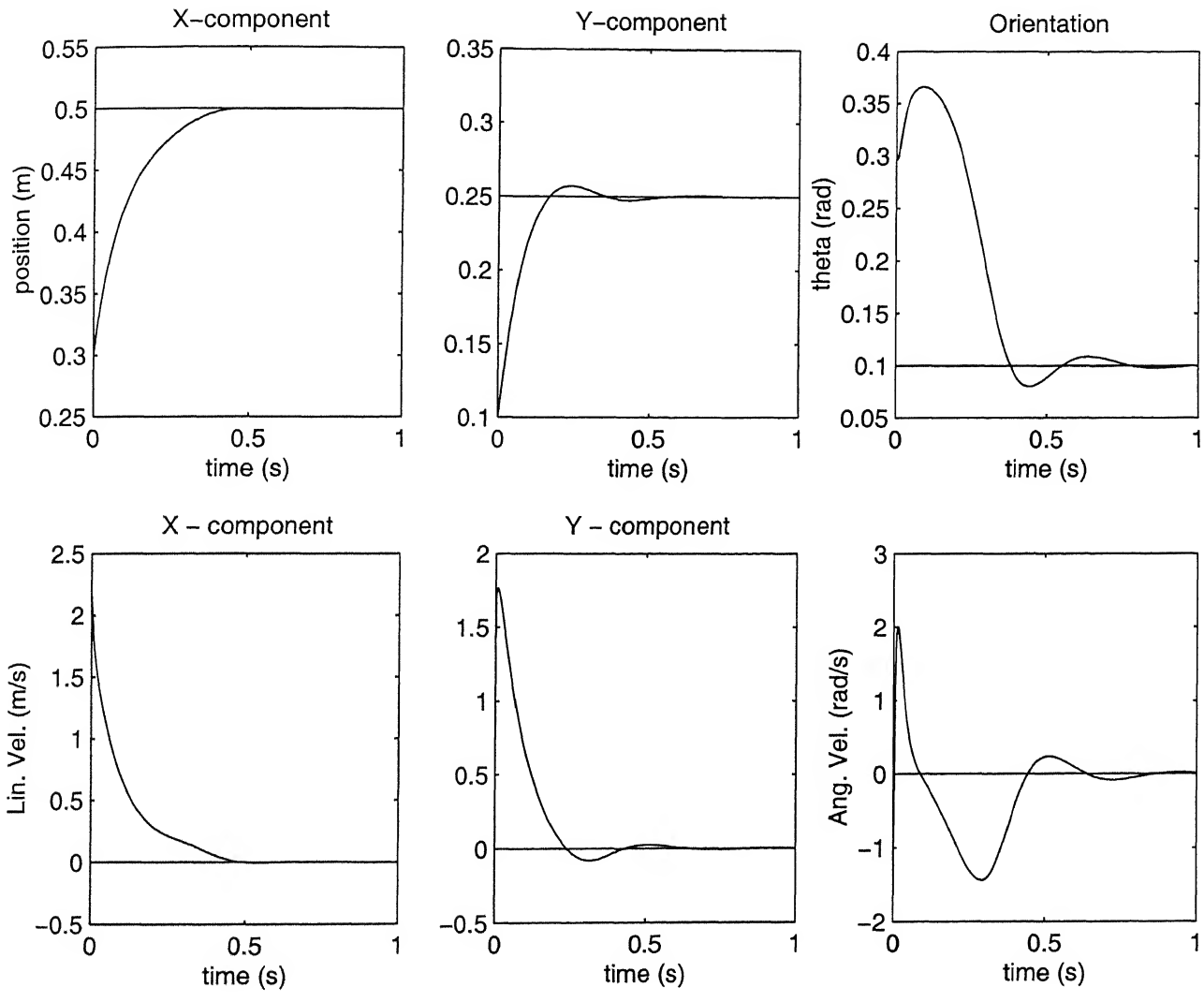


Figure 2.9: Joint-space control of a 8-Bar Manipulator.

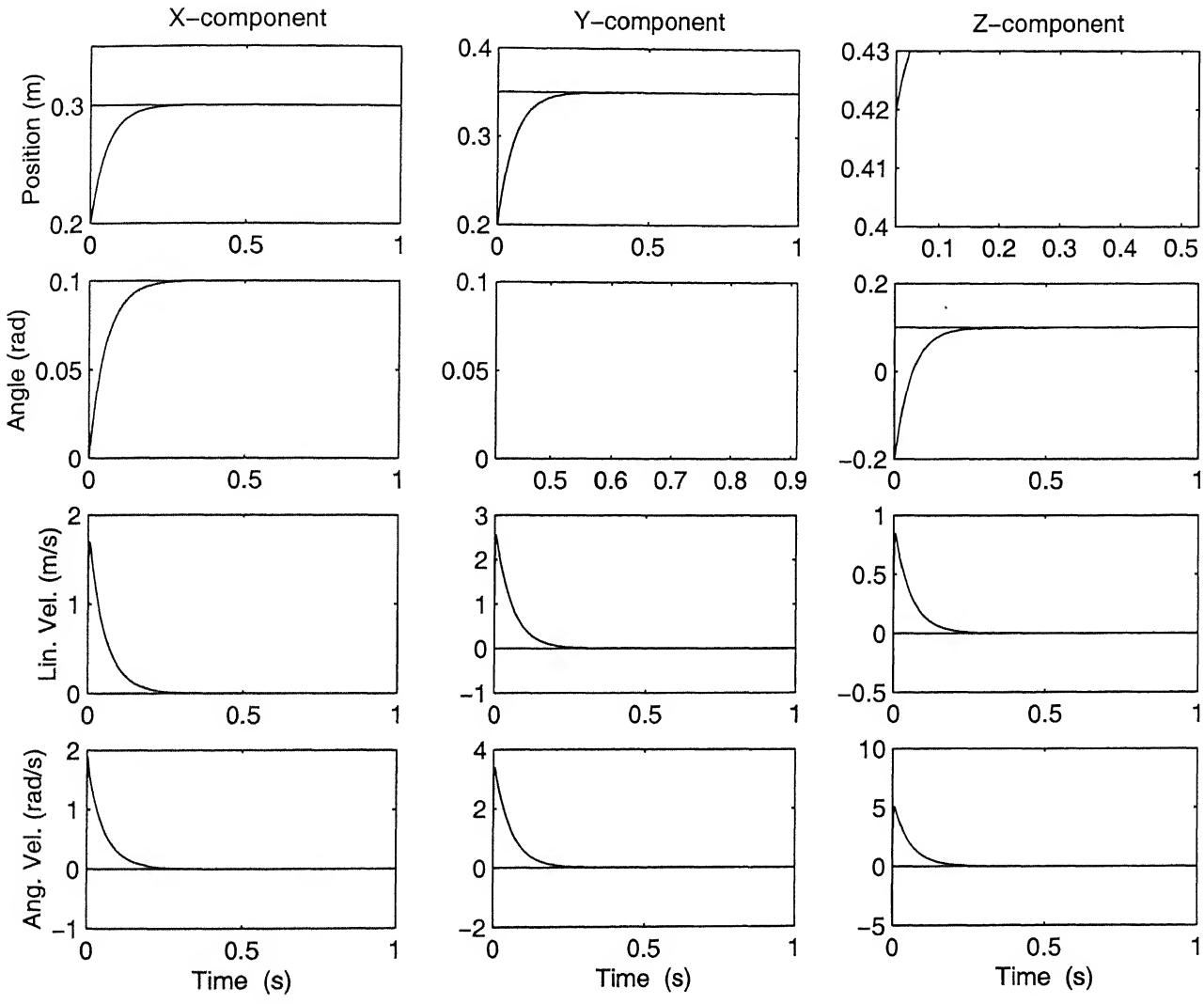


Figure 2.10: Model-based Regulation of a 3-PRPS Manipulator.

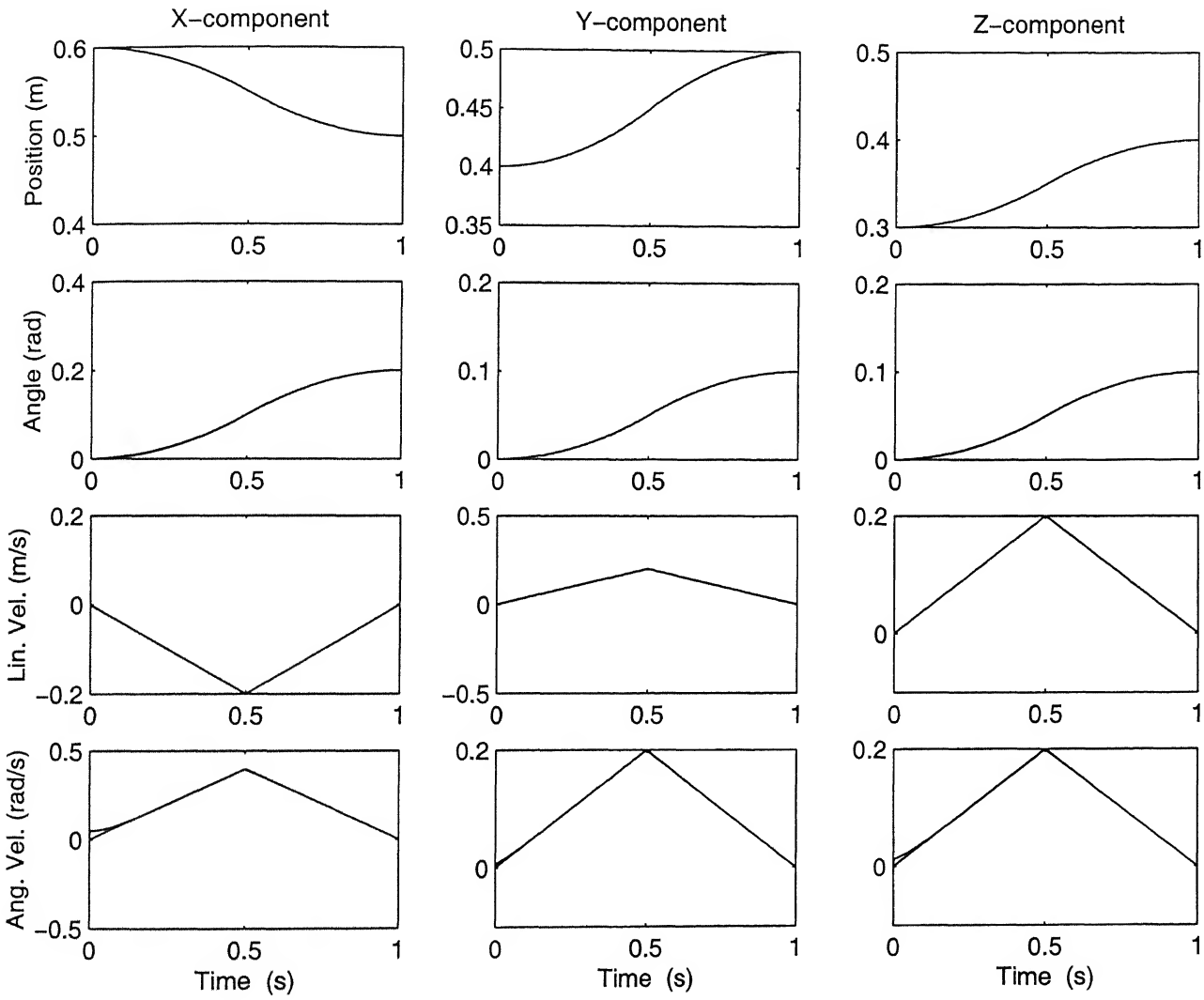


Figure 2.11: Model-based Tracking of a 3-PRPS Manipulator.

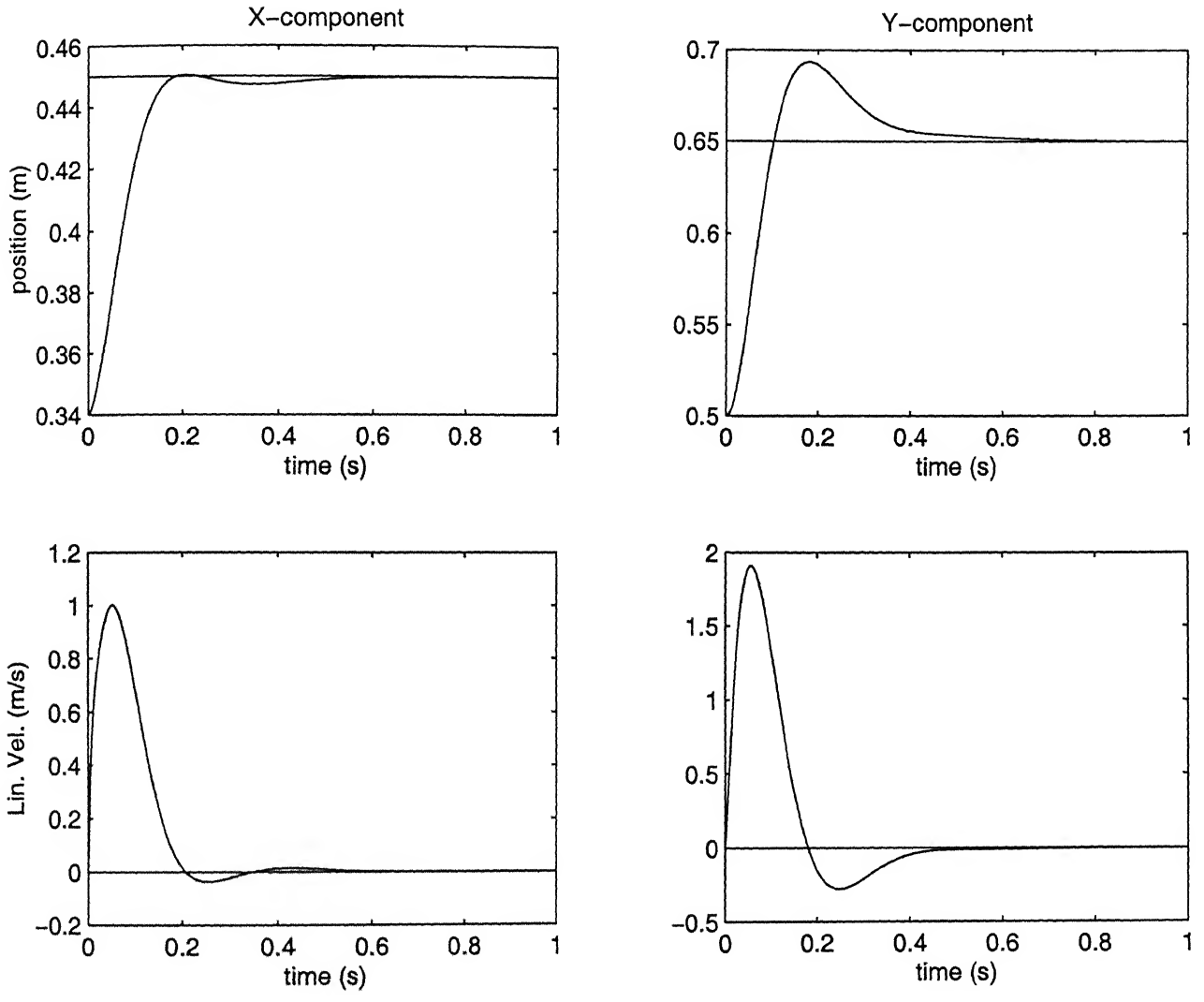


Figure 2.12: Regulation of a 5-Bar Manipulator using Optimal Control.

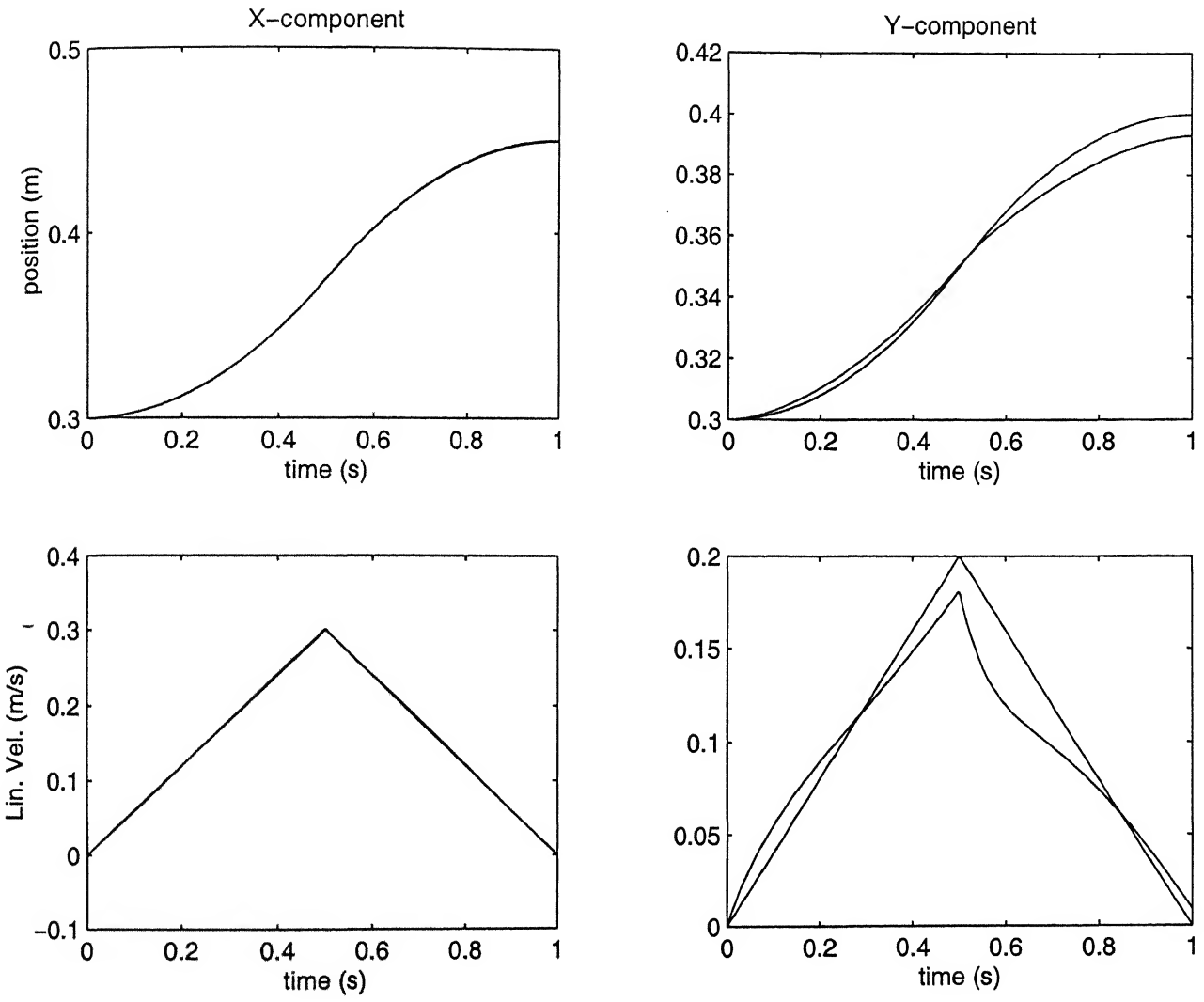


Figure 2.13: Tracking of a 5-Bar Manipulator using Optimal Control.

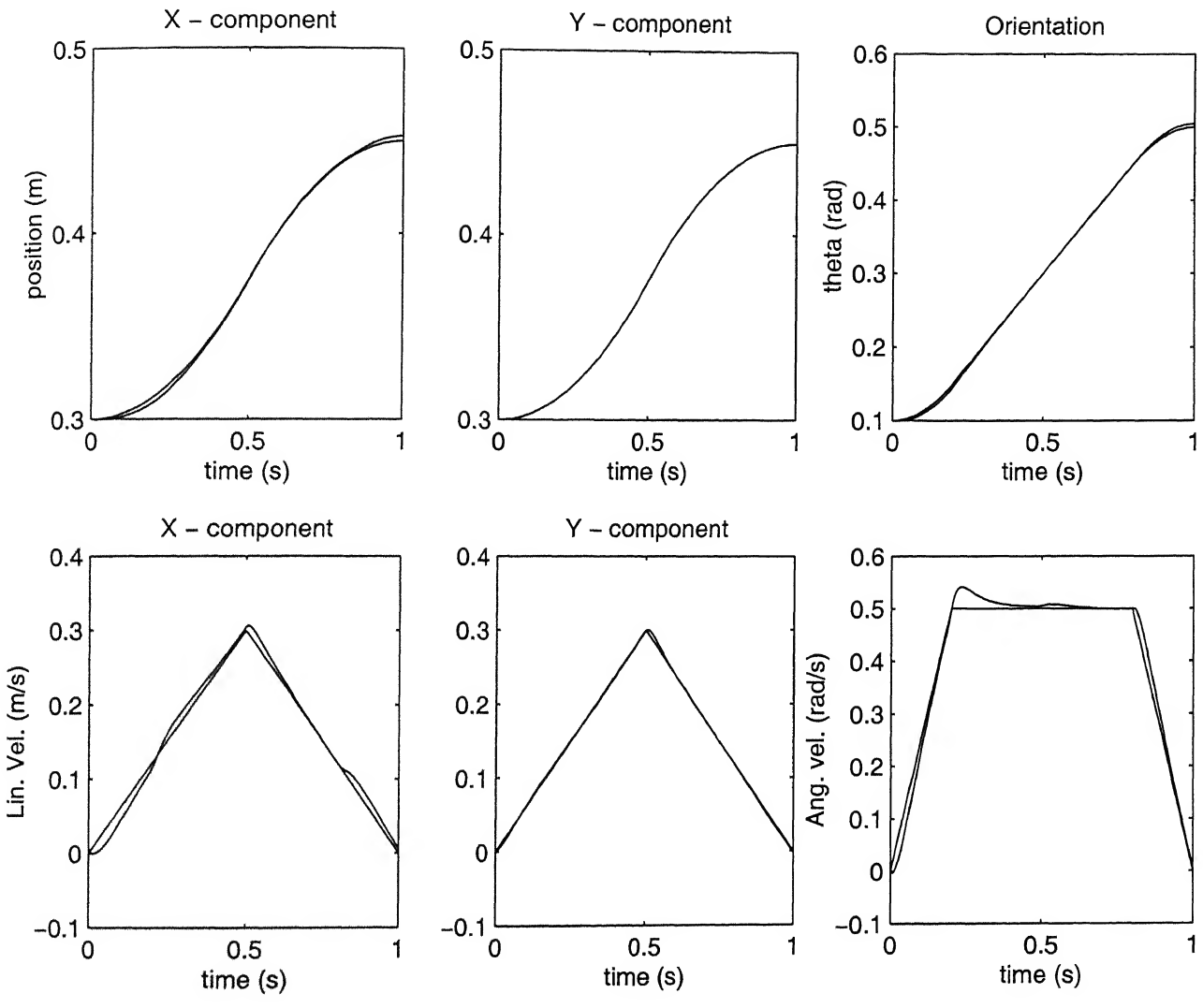


Figure 2.14: Tracking of a 8-Bar Manipulator using PD Control.

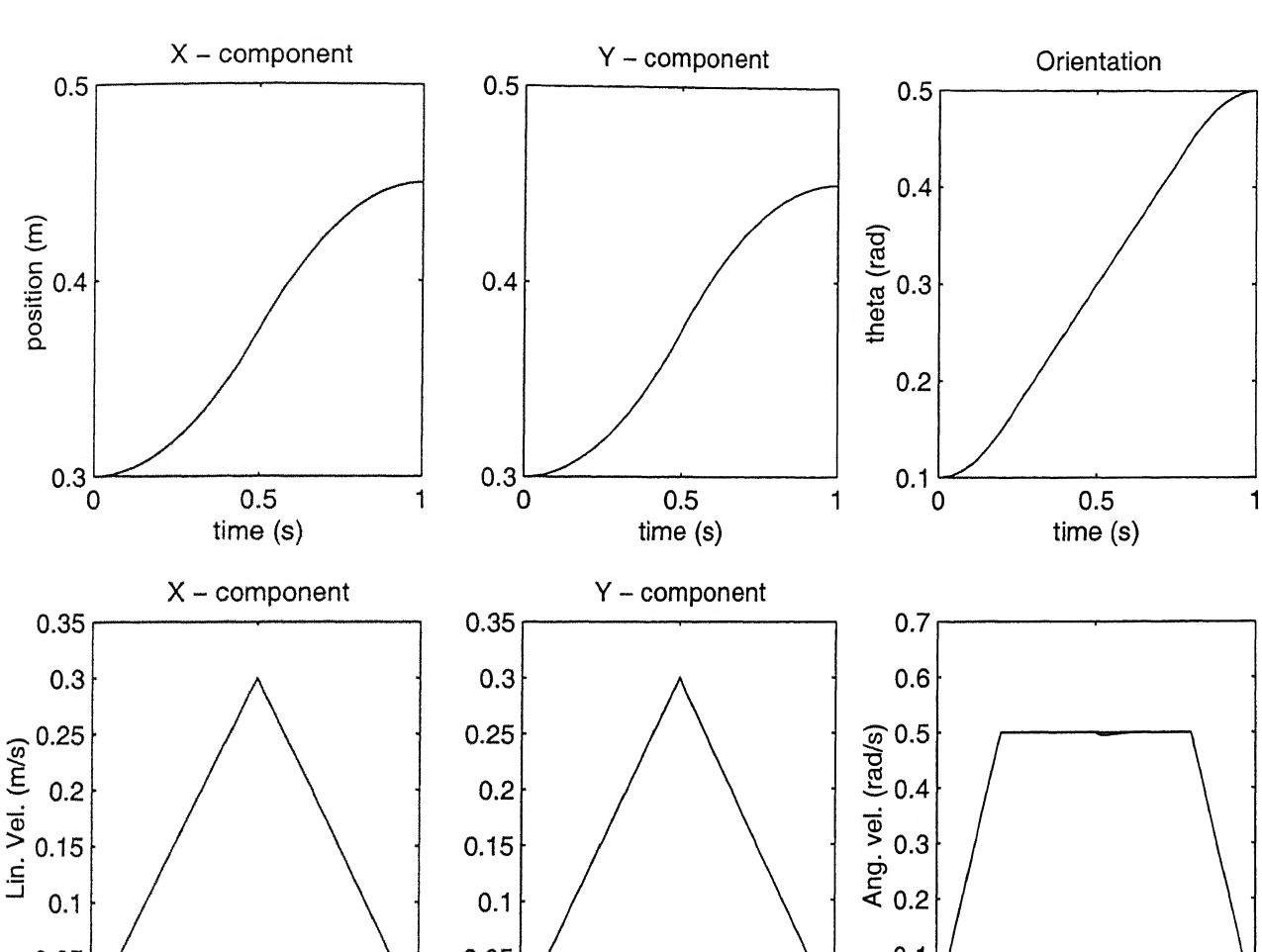


Figure 2.15: Tracking of a 8-Bar Manipulator using Model-based Control.

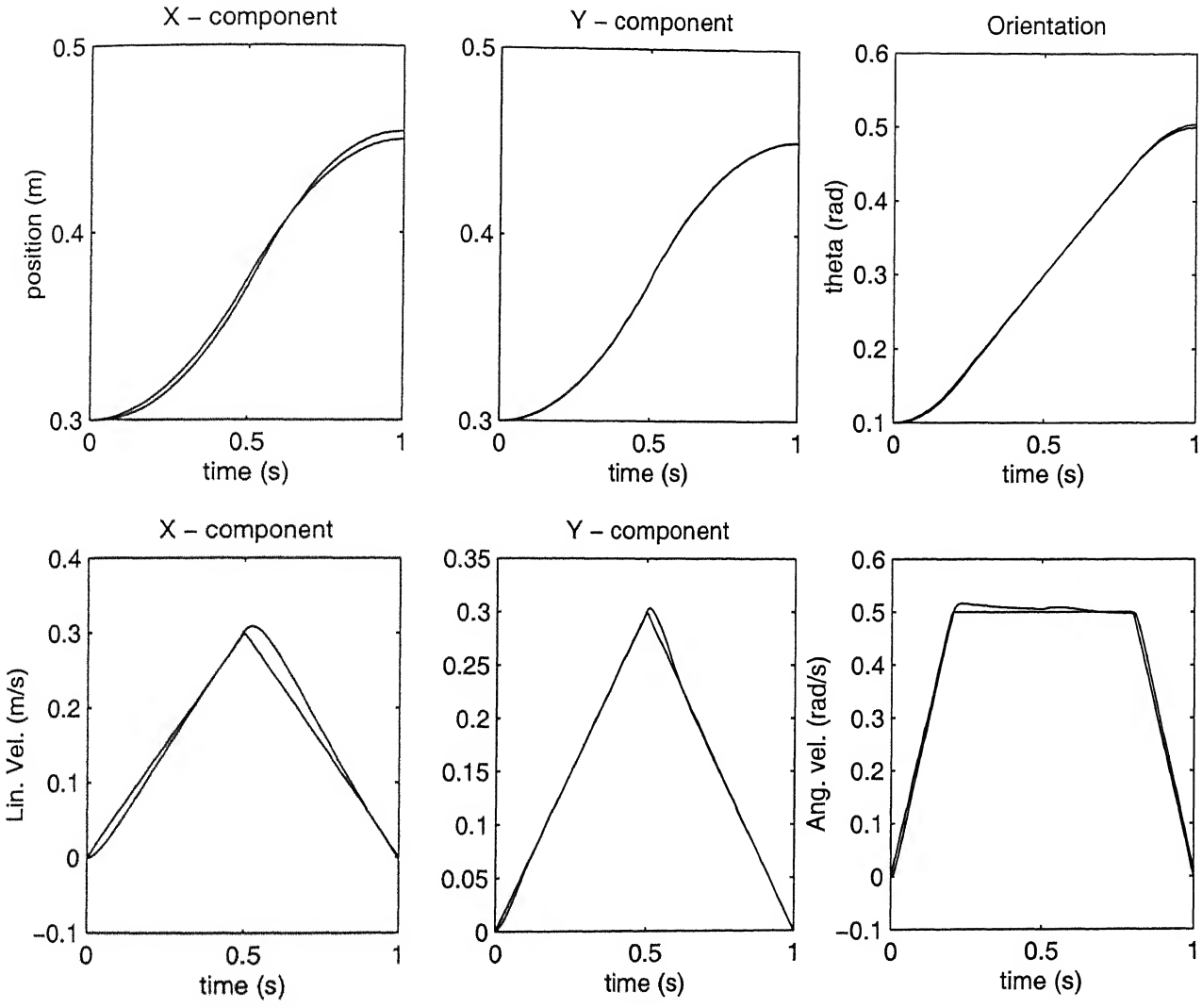


Figure 2.16: Tracking of a 8-Bar Manipulator using Optimal Control.

Chapter 3

Singularity-Free Path-Planning of Parallel Manipulators

3.1 Introduction

The characteristics in which serial and parallel manipulators differ, were studied by Waldron and Hunt [33] in the theory of series-parallel duality. One of the significant contrasts between series and parallel manipulators is the nature of singularities. Serial manipulators have only kinematic singularities, i.e., configurations at which motion of the end-effector in certain directions cannot be obtained by motion of the actuated joints. These singularities are also called workspace singularities as they are encountered at the workspace boundaries. While kinematic singularities are possible in parallel manipulators, the preponderant type of singularities are force singularities. In case of force singularities, the loads acting along some directions on the end-effector cannot be supported by the actuator/joint forces and/or moments. In case of parallel manipulators, workspace singularities appear from the leg limits or the joint limits depending on the type of manipulator. However, parallel manipulators do possess force singularities where they lose some degree of constraint, i.e., gain some degrees of freedom, and become uncontrollable. The analytical condition for force singularities is determined to be the singularities of the force-transformation matrices which map the actuator/joint forces/torques at the legs/joints to the output forces/moments at the end-effector.

Due to differences in the areas of application of serial and parallel manipulators, the nature of the path-planning problems and their requirements differ between the two cases. For serial manipulators, in a path-planning problem, the objective is to move the end-effector from one configuration to another such that, the path remains within the workspace and there is no collision with the obstacles at any point. In case of parallel manipulators, the associated question of practical importance is that of avoiding singularities in the planning of a path for execution of a task. The problem of singularity avoidance can be posed in the form of a path-planning problem where it is necessary to avoid singularities, whose information is available in the form of implicit functional relationship among the task-space coordinates.

For serial manipulators, there exists a large number of methods for solving the path-planning problem. Some of those methods are *roadmap*, *cell decomposition* and *potential field*. The roadmap and cell decomposition methods reduce the problem of finding a path to that of searching a graph [34] by analyzing the connectivity of free-space. In contrast, potential field methods search a much larger graph representing the adjacency among the configurations contained in a grid thrown over the configuration space. At each step, the robot is moved from one configuration to another, based on the potential gradient. The gradient depends on the contents of the configuration space in the neighbourhood of the current configuration of the robot. The potential field is generated in such a manner, that the potential near an obstacle or the workspace boundary is more than that away from an obstacle or far from the workspace boundary. Potential field was originally developed as an online collision avoidance approach, applicable when the robot does not have a prior model of the obstacles, but senses them during motion execution [35]. Emphasis was put on real-time efficiency, rather than on guaranteeing the attaintment of the goal. Another approach to path-planning using the potential field is the Variational approach [36, 37, 38, 39]. This approach consistes of constructing a functional of the path and optimizing this functional over all possible paths. The optimization of the functional can be done using standard variational calculus methods.

While there have been a lot of research work done in the field of path-planning of serial manipulators, research in the field of path-planning of parallel manipulators is much less. Bhattacharya *et al.* [21] have developed a scheme for avoiding singularities of a Stewart platform by restructuring a pre-planned path in the vicinity of a singularity. Dasgupta and Mruthyunjaya [8] have developed an algorithm for planning a well-conditioned path between two end-points for a Stewart platform manipulator. The algorithm also indicates the non-existence of a valid path. Innocenti and Castelli [23] showed that a singularity-free configuration change is possible for parallel manipulators. In this paper, we implement the Variational approach for singularity-free path-planning of parallel manipulators.

In the next section, the analytical criterion for singularities of different types of parallel manipulators have been discussed. In section 3, the problem of singularity-free path planning has been posed. In section 4 the path-planning approach is discussed. In section 5, we present a few examples where the current approach has been applied.

3.2 Singularities and Ill-Conditioning of Parallel Manipulators

As the present work attempts to plan singularity-free paths in the task-space, it is necessary to describe the criterion for singularity in terms of the task-space coordinates.

3.2.1 Static Singularity

When $\mathbf{H}$ is singular, the transformation matrix is degenerate and some loads on the platform cannot be supported by the actuator forces ($\mathbf{F}$) leading to loss of constraint, i.e., gain of freedom by the platform, which causes static singularity. Thus, the criterion of static singularity can be expressed in identity with the singularity of the $\mathbf{H}$ matrix, i.e.,

$$\det[\mathbf{H}(\mathbf{X})] = 0. \tag{3.1}$$

3.2.2 5-Bar Mechanism with Prismatic Actuations

Figure 2.1 shows a 5-bar mechanism with prismatic actuations. The force transformation matrix $\mathbf{H}$, derived in Chapter 2, is given as:

$$\mathbf{H} = \begin{bmatrix} s_{1x} & s_{2x} \\ s_{1y} & s_{2y} \end{bmatrix}$$

where $\mathbf{s}_1$ and $\mathbf{s}_2$ are the unit vectors along the direction of the prismatic joints. This can also be written as:

$$\mathbf{H} = \begin{bmatrix} \cos\phi_1 & \cos\phi_2 \\ \sin\phi_1 & \sin\phi_2 \end{bmatrix} \tag{3.2}$$

where ϕ denotes the angle made by the leg with the X-axis. The force-singularity condition, $\det[\mathbf{H}] = 0$, gives $\sin(\phi_2 - \phi_1) = 0$, or, $\phi_2 = \phi_1 + n\pi$, where $n = 0, \pm 1$. This implies, that the singularity occurs when the end-effector lies on the line connecting the base-points. For these manipulators, kinematic singularities (loss of degree of freedom) occur when maximum or minimum leg limit occurs in any one or both of the legs.

3.2.3 5-Bar Mechanism with Revolute Actuations

Figure 2.2 shows a 5-bar mechanism with revolute actuations. The force transformation matrix $\mathbf{H}$ is given as:

$$\mathbf{H} = \begin{bmatrix} \frac{\mathbf{d}_1}{l_1 \sin(\phi_1 - \theta_1)} & \frac{\mathbf{d}_2}{l_2 \sin(\phi_2 - \theta_2)} \end{bmatrix} \tag{3.3}$$

The singularity condition, namely '$\det[\mathbf{H}] = 0$', reduces to:

$$d_{1x} \times d_{2y} - d_{1y} \times d_{2x} = 0 \tag{3.4}$$

In the Cartesian (output) space the above condition represents a curve. For these manipulators kinematic singularities occur when:

$$\phi_i = \theta_i$$

3.2.4 8-Bar Manipulator with Prismatic Actuations

Figure 2.3 shows a 3-DOF parallel manipulator with prismatic actuations. The force transformation matrix is given by:

$$\mathbf{H} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 \\ \mathbf{q}_1 \times \mathbf{s}_1 & \mathbf{q}_2 \times \mathbf{s}_2 & \mathbf{q}_3 \times \mathbf{s}_3 \end{bmatrix} \tag{3.5}$$

From the kinematic analysis of this mechanism, we have:

$$\mathbf{S}_i = \mathbf{t} + \Re\mathbf{p}_i - \mathbf{b}_i$$

where $\Re$ is the rotation matrix. Using the above relationship after simplification, we get:

$$\mathbf{H} \sim \begin{bmatrix} \mathbf{t} + \Re\mathbf{p}_1 - \mathbf{b}_1 & \Re(\mathbf{p}_2 - \mathbf{p}_1) - (\mathbf{b}_2 - \mathbf{b}_1) & \Re(\mathbf{p}_3 - \mathbf{p}_1) - (\mathbf{b}_3 - \mathbf{b}_1) \\ \mathbf{b}_1 \times (\mathbf{t} + \Re\mathbf{p}_1) & (\mathbf{b}_2 - \mathbf{b}_1) \times \mathbf{t} + \mathbf{b}_2 \times \Re\mathbf{p}_2 - \mathbf{b}_1 \times \Re\mathbf{p}_1 & (\mathbf{b}_3 - \mathbf{b}_1) \times \mathbf{t} + \mathbf{b}_3 \times \Re\mathbf{p}_3 - \mathbf{b}_1 \times \Re \end{bmatrix}$$

For a constant orientation of the output platform, the matrix $\Re$ will be constant. In such a case, the matrix $\mathbf{H}$ can be furthur simplified and written in the form:

$$\mathbf{H} \sim \begin{bmatrix} x^1 & c & c \\ x^1 & c & c \\ x^1 & x^1 & x^1 \end{bmatrix}$$

where the linear and constant terms are denoted as x^1 and c, respectively. On expanding and equating $\det[\mathbf{H}] = 0$, we get a quadratic singularity curve. This is called degeneracy of rank 2. This singularity occurs when the legs become concurrent and the manipulator is unable to withstand external moments on the platform. We will also encounter singularity when the legs become parallel and the manipulator is unable to resist any externally applied force on the platform in a direction perpendicular to the leg.

3.2.5 8-Bar Manipulator with Revolute Actuations

Figure 2.4 shows a 3-DOF parallel manipulator with revolute actuations. The force singularity condition occurs when the links connecting the first link to the platform are either parallel or concurrent. When the links are parallel, the mechanism gains a translational degree of freedom in a direction normal to the leg vector and when the legs are concurrent the mechanism can rotate about the point of concurrency even when the actuators are locked. For this mechanism the force transformation matrix is given by:

$$\mathbf{H} = \begin{bmatrix} \frac{\mathbf{d}_1}{l_1 \sin(\phi_1 - \theta_1)} & \frac{\mathbf{d}_2}{l_2 \sin(\phi_2 - \theta_2)} & \frac{\mathbf{d}_3}{l_3 \sin(\phi_3 - \theta_3)} \\ \frac{\mathbf{q}_1 \times \mathbf{d}_1}{l_1 \sin(\phi_1 - \theta_1)} & \frac{\mathbf{q}_2 \times \mathbf{d}_2}{l_2 \sin(\phi_2 - \theta_2)} & \frac{\mathbf{q}_3 \times \mathbf{d}_3}{l_3 \sin(\phi_3 - \theta_3)} \end{bmatrix} \tag{3.6}$$

From the singularity condition, the nature of the curves cannot be determined analytically as even under constant orientation, the matrix $\mathbf{H}$ contains non-linear trigonometric terms.

3.2.6 3-PRPS Hybrid Manipulator

Figure 2.5 shows a 3-PRPS hybrid manipulator. For this case, the force transfromation matrix has 2 components – due to the actuator forces and due to the constraint forces. It can be written in a symbolic form as:

$$\mathbf{H} = [\mathbf{H}_k \quad \mathbf{H}_s]$$

or in an expanded form as:

$$\mathbf{H} = \begin{bmatrix} \mathbf{k}_1 & \mathbf{k}_2 & \mathbf{k}_3 & \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 \\ \mathbf{q}_1 \times \mathbf{k}_1 & \mathbf{q}_2 \times \mathbf{k}_2 & \mathbf{q}_3 \times \mathbf{k}_3 & \mathbf{q}_1 \times \mathbf{s}_1 & \mathbf{q}_2 \times \mathbf{s}_2 & \mathbf{q}_3 \times \mathbf{s}_3 \end{bmatrix} \tag{3.7}$$

The matrix $\mathbf{H}_s$ corresponds to the leg actuations and $\mathbf{H}_k$ corresponds to the forces acting along the prismatic joints at the bases. The manipulator reaches singular configurations when either the actuation or constraint forces become infinite or the manipulator is unable to withstand a certain combination of external forces and moments with the given actuator and constraint forces. If the rank of the matrix $\mathbf{H}_s$ becomes less than 2, the condition is similar to "degeneracy of rank 2", where the legs become parallel or concurrent.

3.2.7 Stewart Platform Manipulator

Figure 3.1 shows a Stewart Platform Manipulator. This manipulator has one prismatic actuation in each leg. The force transformation matrix of a Stewart Platform Manipulator can be shown to be:

$$\mathbf{H} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 & \mathbf{s}_4 & \mathbf{s}_5 & \mathbf{s}_6 \\ \mathbf{q}_1 \times \mathbf{s}_1 & \mathbf{q}_2 \times \mathbf{s}_2 & \mathbf{q}_3 \times \mathbf{s}_3 & \mathbf{q}_4 \times \mathbf{s}_4 & \mathbf{q}_5 \times \mathbf{s}_5 & \mathbf{q}_6 \times \mathbf{s}_6 \end{bmatrix}. \tag{3.8}$$

The singularity manifold consists of hypersurfaces in the six-dimensional task-space.

3.2.8 Ill-Conditioning

An important concept associated with singularity is that of ill-conditioning. When the manipulator is near rather than exactly at a singularity, enormous amounts of forces at the actuators may be required to support even reasonable magnitudes of loads at the platform, and from a practical standpoint control is neverthless lost. This situation is called ill-conditioning and can be identified by very small values of det[**H**]. However, while the determinant serves well in givng the analytical criterion of singularity, it is not considered a good measure of ill-conditioning. A better measure of ill-conditioning is provided by the condition number of the matrix **H**. The condition number becomes infinite at a singularity. In practice, the ill-conditioning of the manipulator can be measured by the condition number:

$$\kappa(\mathbf{X}) = \text{cond}[\mathbf{W}\mathbf{H}(\mathbf{X})] \tag{3.9}$$

where **W** is a 6×6 weight matrix used to compensate for dimensionally heterogeneous character of the force-transformation matrix **H**.

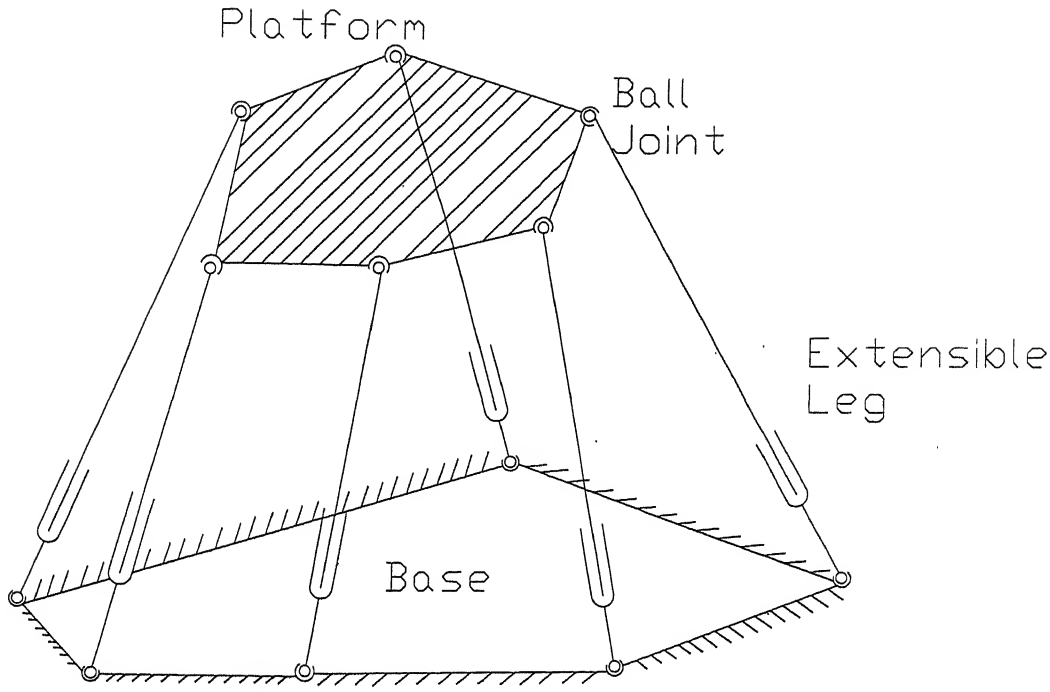


Figure 3.1: Stewart Platform Manipulator.

3.3 Singularity-free Path Planning

In this section we formulate the path planning problem for parallel manipulators. Here planning of a path as a function of a parameter t is considered with the understanding that at the trajectory planning stage the pre-planned path can be reparametrized with respect to time. The primary objective of the path-planning strategy is to find a path completely within the workspace connecting the initial configuration ($\mathbf{X}_i$) to the final configuration ($\mathbf{X}_f$), i.e. to find the function:

$$\mathbf{X} = \mathbf{X}(t) \quad \text{for} \quad 0 \leq t \leq 1, \tag{3.10}$$

where

$$\mathbf{X}(0) = \mathbf{X}_i \quad \text{and} \quad \mathbf{X}(1) = \mathbf{X}_f,$$

such that

$$L^i_{min} \leq L_i(t) \leq L^i_{max} \quad \text{for} \quad i = 1 \text{ to N} \quad \text{and} \quad 0 \leq t \leq 1. \tag{3.11}$$

where N is the total number of actuators. In addition, for the path to be singularity-free and well-conditioned everywhere, the determinant of the matrix $\mathbf{H}$ must be of the same sign along the entire path and the condition number must always be limited by:

$$\kappa[\mathbf{X}(t)] \leq \kappa_{lim} \tag{3.12}$$

where κ_{lim} is an allowable condition number to be prescribed on the basis of expected end-effector load and actuator capacities.

3.4 Path Planning by Variational Approach

In Variational approach we construct a functional L given by:

$$L = K(\mathbf{X}, \dot{\mathbf{X}}) - P(\mathbf{X}) \tag{3.13}$$

where $K(\mathbf{X}, \dot{\mathbf{X}})$ is the Kinetic Energy term which tries to keep the path short, and is given by:

$$K(\mathbf{X}, \dot{\mathbf{X}}) = \frac{1}{2}\dot{\mathbf{X}}^T \mathbf{M} \dot{\mathbf{X}}. \tag{3.14}$$

The matrix $\mathbf{M}$ is used for scaling the different components of the vector $\mathbf{X}$. $P(\mathbf{X})$ is the Potential Energy term which ensures that the path remains within the workspace (i.e., the leg-limits are not violated) and the path is singularity-free in nature. The expression for this term is given by:

$$P(\mathbf{X}) = \sum_{i=1}^{N} P_i(\mathbf{X}) + P_{det}(\mathbf{X}), \tag{3.15}$$

where $P_i(\mathbf{X})$ accounts for the potential due to the leg-limits imposed by the i-th leg, and $P_{det}(\mathbf{X})$ is the potential due to the ill-conditioning of the force-transformation matrix $\mathbf{H}$. The i-th leg-length constraints are given by:

$$\begin{aligned} \Delta L^i_{low} &= L^i_{min} - L_i \leq 0 \\ \Delta L^i_{high} &= L_i - L^i_{max} \leq 0 \end{aligned}$$

The potentials associated with these constraints are:

$$P^i_{low} = c_1, \text{ if } \Delta L^i_{low} > 0 \tag{3.16}$$

$$P^i_{low} = c_1 e^{c_2 \Delta L^i_{low}} \text{ otherwise.} \tag{3.17}$$

$$P^i_{high} = c_1, \text{ if } \Delta L^i_{high} > 0 \tag{3.18}$$

$$P^i_{high} = c_1 e^{c_2 \Delta L^i_{high}} \text{ otherwise.} \tag{3.19}$$

The constant c_1 is chosen as a very large number, so that the potential due to leg-limit violation is very high. The potential at surrounding points decrease gradually till it reduces to zero at points within the workspace. The rapidity of this decrement is controlled by the constant c_2, which is a positive number. The potential of i-th leg is the sum of the potentials P^i_{low} and P^i_{high}. Thus,

$$P_i = P^i_{low} + P^i_{high} \tag{3.20}$$

The potential due to the ill-conditioning of the force-transformation matrix $\mathbf{H}$ is given by the equation:

$$P_{det} = c_3, \text{ if } \text{cond}(\mathbf{H}) > \kappa_{lim} \tag{3.21}$$

$$P_{det} = c_3 e^{c_4[\text{cond}(\mathbf{H})-\kappa_{lim}]} \text{ if } \text{cond}(\mathbf{H}) \leq \kappa_{lim} \tag{3.22}$$

The constant c_3 is chosen as a very large number, so that the potential near a singularity is very high. The potential at surrounding points decrease gradually till it reduces to zero at points within the workspace which are far from any singularities. The rapidity of this decrement is controlled by the constant c_4, which is a positive number. By minimising the integral $\int_0^1 Ldt$ we get the Euler-Lagrange equations in the form:

$$\mathbf{M}\ddot{\mathbf{X}} = -\frac{\partial P}{\partial \mathbf{X}} \tag{3.23}$$

Solution of this equation is a two-point boundary-value problem. Unlike initial-value problems, boundary-value problems may or may not have a solution, may have finite or infinite number of solutions. In our case, we have to choose an appropriate value of $\dot{\mathbf{X}}(0)$, so that by integrating this system of equations we would get the required path, which will remain inside the workspace and will be singularity-free in nature. There are two distinct classes of numerical methods for solving two-point boundary-value problems. These are *Shooting methods* and *Relaxation methods*. In the *Shooting method* we choose values for all the dependent variables at one boundary which are consistent with the boundary conditions on that boundary, but otherwise are arranged to depend on arbitrary free parameters whose values are guessed "randomly". The ODE's are then integrated by initial-value methods arriving at the other boundary. In general, there will be discrepancies in the boundary conditions at the other end. Depending on the discrepancies, we find the adjustment of the free parameters at the starting point which zeros the discrepancies at the other boundary. *Relaxation methods* use a different approach. Differential equations are replaced by finite-difference equations on a mesh of points that covers the range of integration. A trial solution consists of values for the dependent variables at each mesh-point, *not* satisfying the finte-difference equations, nor necessarily even satisfying the boundary conditions. The iteration, now called *relaxation*, consists of adjusting all the values on the mesh so as to bring them in closer agreement with the finite-difference equations and the boundary conditions. *Relaxation* works better than *Shooting* in most cases. But for it to work efficiently, good initial guesses must be provided.

3.5 Results and Discussion

The path planning algorithm presented in the last section has been implemented in MATLAB routines for generating singularity-free paths for different classes of parallel manipulators. The program developed in the present work uses the MATLAB routines "bvpinit", "bvp4c" and

"bvpval". The MATLAB routine "bvp4c" solves two-point boundary value problems for ordinary differential equations. "bvp4c" is a finite difference code which implements the 3 stage Lobatto Illa formula. This is a collocation formula providing a C^1-continuous solution which is fourth-order accurate uniformly in the interval of integration. Mesh selection and error control are based on the residual of the continuous solution. "bvpinit" is a function used to obtain a proper guess for the "bvp4c" function. The function "bvpval" uses the output of "bvp4c" to evaluate the solution at specific points. The weight matrix $\mathbf{W}$ in equation (3.9) has been chosen as identity matrix with the assumption that the order of manipulator dimensions (for example, distances of base-points from the base-frame origin) serves as reasonable normalizing distance and gives a proper scaling between forces and moments for comparison. However, different weight matrices can be chosen depending on particular applications.

For κ_{lim}, the maximum allowable condition number, a value of $\kappa_{lim} = 100$ is chosen by default if the initial and final poses have sufficiently low condition numbers; otherwise a value of 1.5 times the greater of the two is selected as κ_{lim}. Thus,

$$\kappa_{lim} = \max[100, 1.5\kappa(\mathbf{X}_i), 1.5\kappa(\mathbf{X}_f)].$$

Again these parameters can be varied from problem to problem.

3.5.1 Example I

In this example, we consider path-planning of a 5-Bar 2-dof planar parallel manipulator. The base-points of the 5-Bar manipulator used in this example are given in Appendix III. Leg-length limits: $L_{min} = 0.3$ m; $L_{max} = 1.0$ m, for both the legs. It is desired to plan a singularity-free path for the manipulator starting from the initial pose

$$\mathbf{X}_i = [-0.6 \quad 0.05]^T \text{ m}$$

to the final pose

$$\mathbf{X}_f = [0.95 \quad 0.1]^T \text{ m}.$$

Figure 3.2 shows the workspace and singularity of the 5-Bar manipulator, and the planned path is also shown. From the figure we see that the planned path has successfully avoided the singularity and has remained within the workspace. Figure 3.3 shows the variation of the condition number and the determinant along the path.

3.5.2 Example II

In this example, we consider path-planning of a 8-Bar 3-dof planar parallel manipulator. The base and platform points of the 8-Bar manipulator used in this example are given in Appendix III.

Leg-length limits: $L_{min} = 0.2$ m; $L_{max} = 1.0$ m, for all the legs. It is desired to plan a singularity-free path for the manipulator starting from the initial pose

$$\mathbf{t}_i = [-0.4 \quad -0.4]^T \text{ m} \quad \text{and} \quad \theta_i = 0.0 \text{ rad},$$

or,

$$\mathbf{X}_i = [-0.4 \quad -0.4 \quad 0.0]^T$$

to the final pose

$$\mathbf{t}_f = [0.4 \quad 0.6]^T \text{ m} \quad \text{and} \quad \theta_f = 0.5 \text{ rad},$$

or,

$$\mathbf{X}_f = [0.4 \quad 0.6 \quad 0.5]^T.$$

Figure 3.4 shows the variation of x, y and θ along the path. Figure 3.5 shows the variation of the condition number and the determinant along the path. Figure 3.6 shows the variation of the leg-lengths along the planned path.

3.5.3 Example III

In this example, we consider path-planning of a 3-PRPS spatial hybrid manipulator. The base-points, platform-points and the fixed-axes of the 3-PRPS manipulator used in this example are given in Appendix III. Leg-length limits: $L_{min} = 0.2$ m; $L_{max} = 1.25$ m, for all the legs. Leg-limits for the prismatic joints along the fixed axes : $D_{min} = 0.1$ m; $D_{max} = 0.5$ m, for all the legs. It is desired to plan a singularity-free path for the manipulator starting from the initial pose

$$\mathbf{t}_i = [0.2 \quad 0.3 \quad 0.3]^T \text{ m} \quad \text{and} \quad \boldsymbol{\theta}_i = [0.0 \quad 0.1 \quad 0.2]^T \text{ rad},$$

or,

$$\mathbf{X}_i = [0.2 \quad 0.3 \quad 0.3 \quad 0.0 \quad 0.1 \quad 0.2]^T$$

to the final pose

$$\mathbf{t}_f = [0.2 \quad 0.5 \quad 0.2]^T \text{ m} \quad \text{and} \quad \boldsymbol{\theta}_f = [0.1 \quad 0.3 \quad 0.4]^T \text{ rad},$$

or,

$$\mathbf{X}_f = [0.2 \quad 0.5 \quad 0.2 \quad 0.1 \quad 0.3 \quad 0.4]^T.$$

Figure 3.7 shows the planned singularity-free path. Figure 3.8 shows the variation of the condition number and the determinant along the path. Figure 3.9 shows the variation of the leg-lengths and the base-joint lengths along the planned path.

3.5.4 Example IV

In this example, we consider path-planning of a Stewart platform manipulator. The base and platform points of the Stewart platform manipulator used in this example are given in Appendix III. Leg-length limits: $L_{min} = 0.2$ m; $L_{max} = 1.5$ m, for all the legs. It is desired to plan a singularity-free path for the manipulator starting from the initial pose

$$\mathbf{t}_i = [0.6 \;\; 0.2 \;\; 0.1]^T \text{ m} \quad \text{and} \quad \boldsymbol{\theta}_i = [0.0 \;\; 0.0 \;\; 0.0]^T \text{ rad},$$

or,

$$\mathbf{X}_i = [0.6 \;\; 0.2 \;\; 0.1 \;\; 0.0 \;\; 0.0 \;\; 0.0]^T$$

to the final pose

$$\mathbf{t}_f = [0.5 \;\; 0.3 \;\; 0.1]^T \text{ m} \quad \text{and} \quad \boldsymbol{\theta}_f = [0.0 \;\; 0.0 \;\; 0.0]^T \text{ rad},$$

or,

$$\mathbf{X}_f = [0.5 \;\; 0.3 \;\; 0.1 \;\; 0.0 \;\; 0.0 \;\; 0.0]^T.$$

Figure 3.10 shows the planned singularity-free path. Figure 3.11 shows the variation of the condition number and the determinant along the path. Figure 3.12 shows the variation of the leg-lengths along the planned path.

3.5.5 Discussion

From all the above cases, one observes that the Variational Approach has not only been successful in finding the singularity-free paths, but also the paths obtained are in some way optimum. That is, within the workspace, at places which are far away from the singularities, the components of the path are nearly straight - that is the obtained paths are quite short. This has been possible due to the inclusion of the kinetic energy term K in our formulation. In all the cases, the leg-length limits are not violated implying the paths are valid paths, i.e., the obtained paths lie within the workspace. Another advantage of the Variational Approach is that it allows additional objectives to be coded in the functional. For cases where the desired end-point is very close to the workspace boundary, the constants are to be chosen suitably so that the potential at the end-point is not very high. It is to be noted, that at the path-planning stage, there is no difference between the 5-Bar manipulators with prismatic joint or revolute joint. The same is true for the 8-Bar manipulators with prismatic joint or revolute joint. For manipulators with revolute joints, the minimum and maximum leg-length limits are to be replaced by the difference and sum of the leg-lengths of the lower and upper legs in each chain.

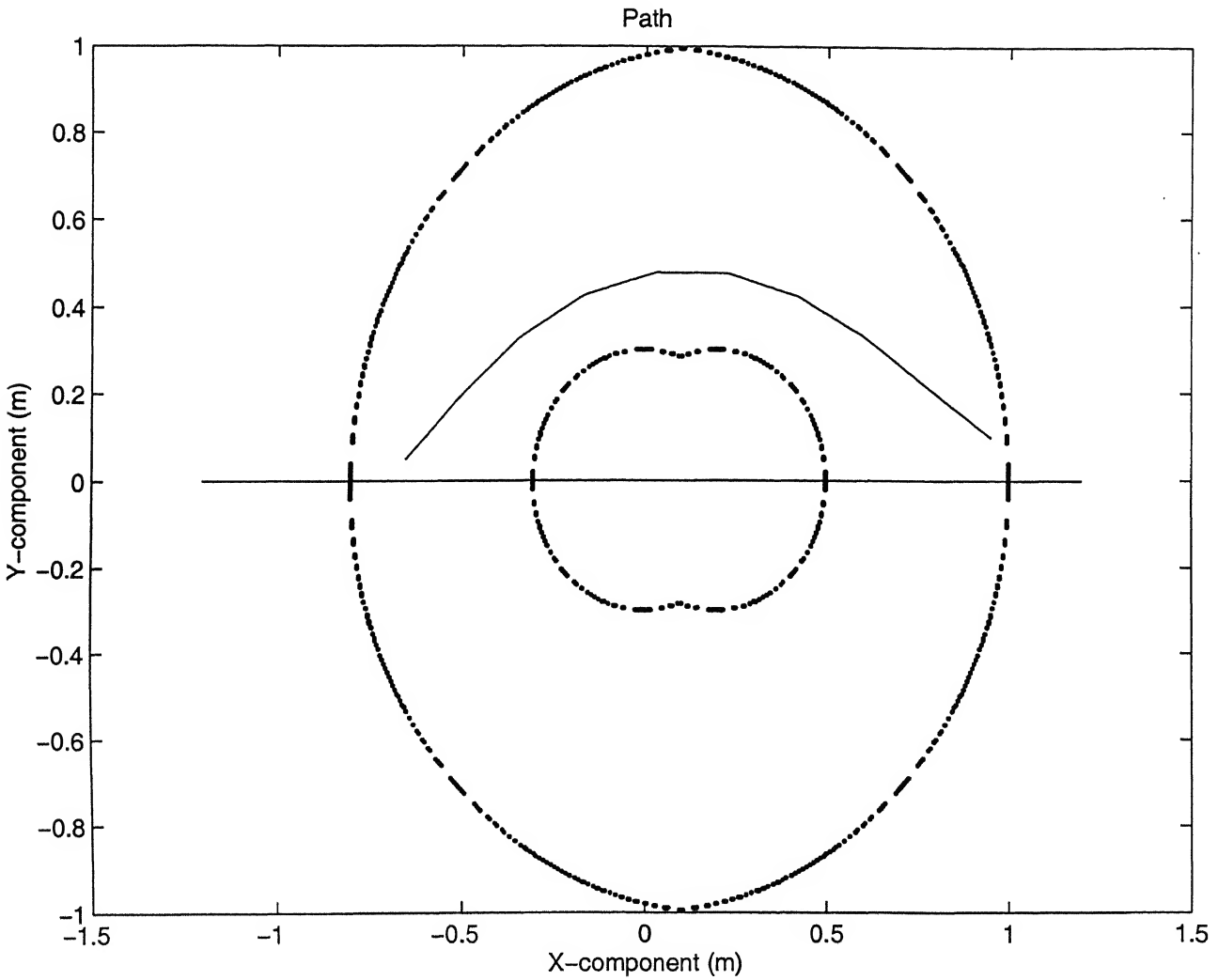
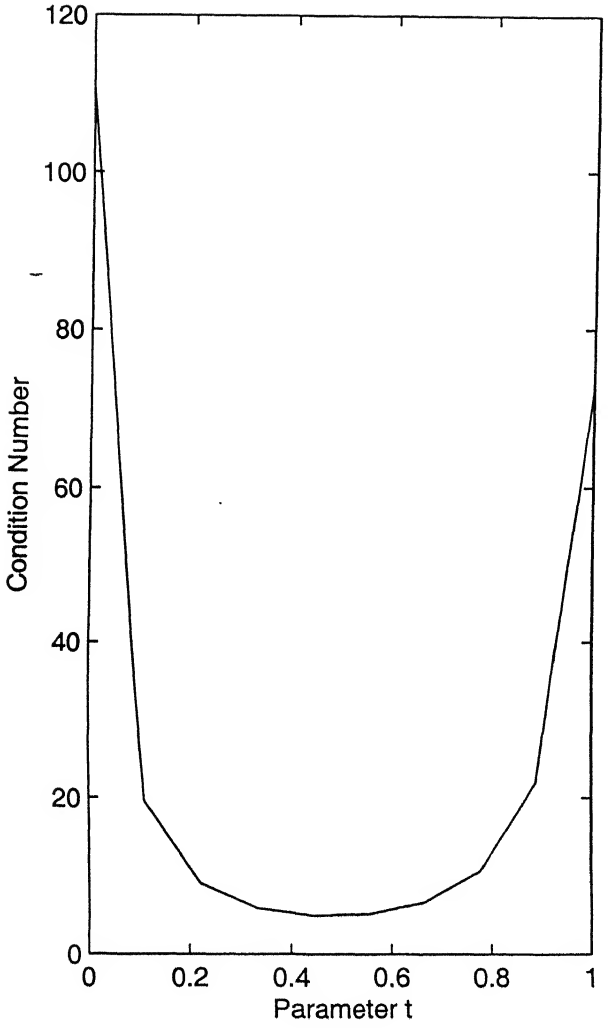


Figure 3.2: Singularity-free planned path for a 5-Bar Manipulator.



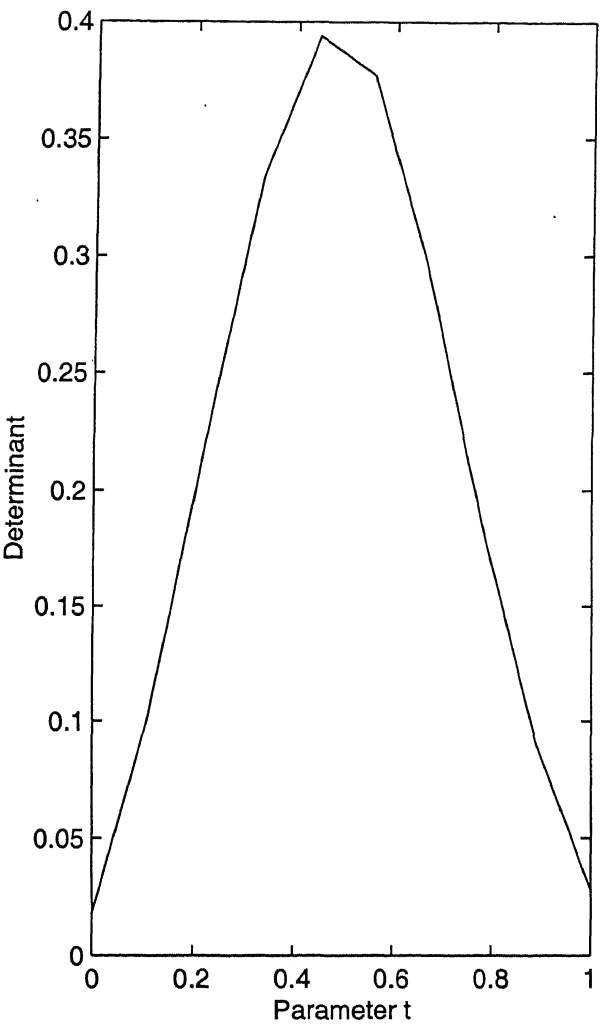


Figure 3.3: Variation of Condition Number and Determinant along the path.

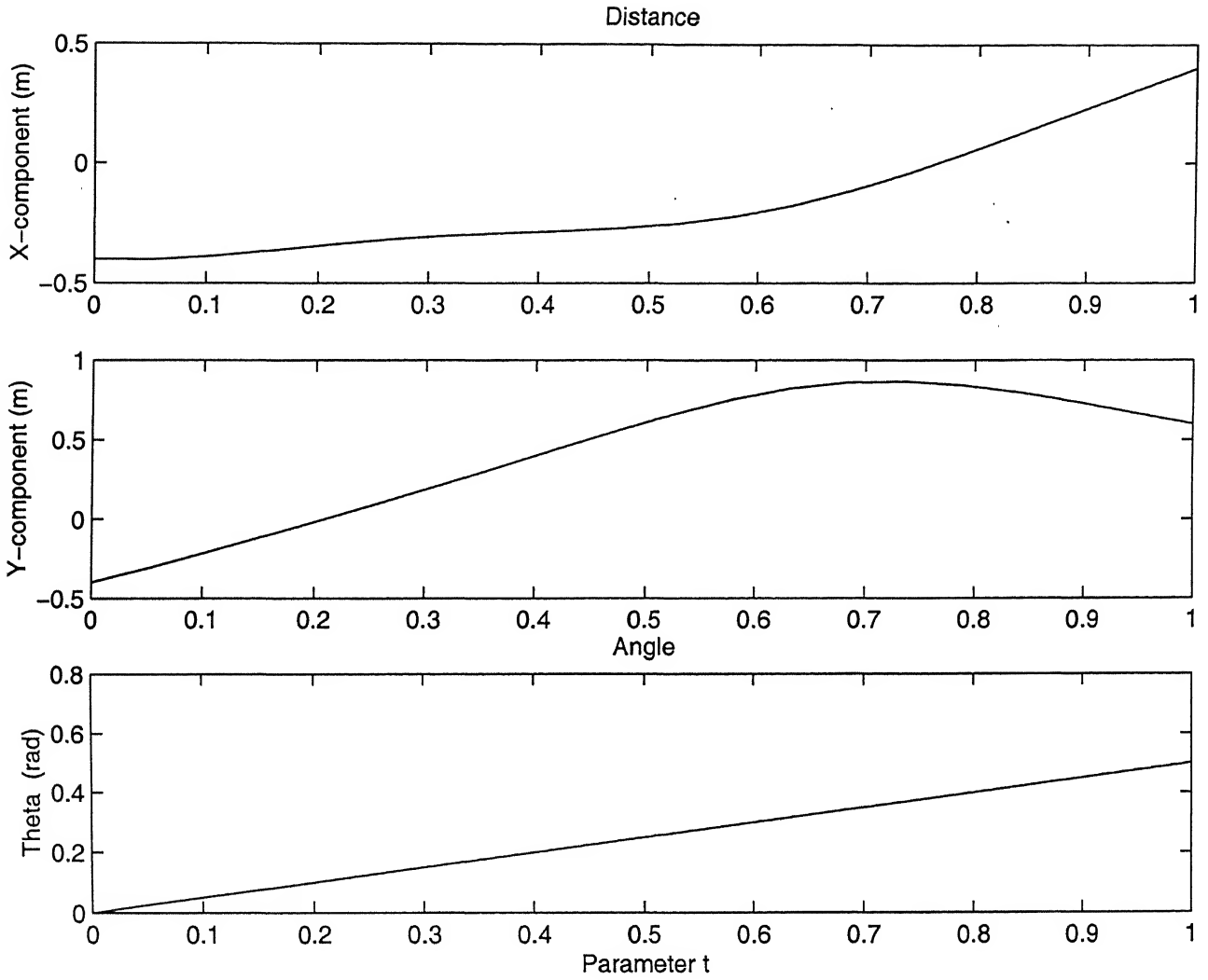
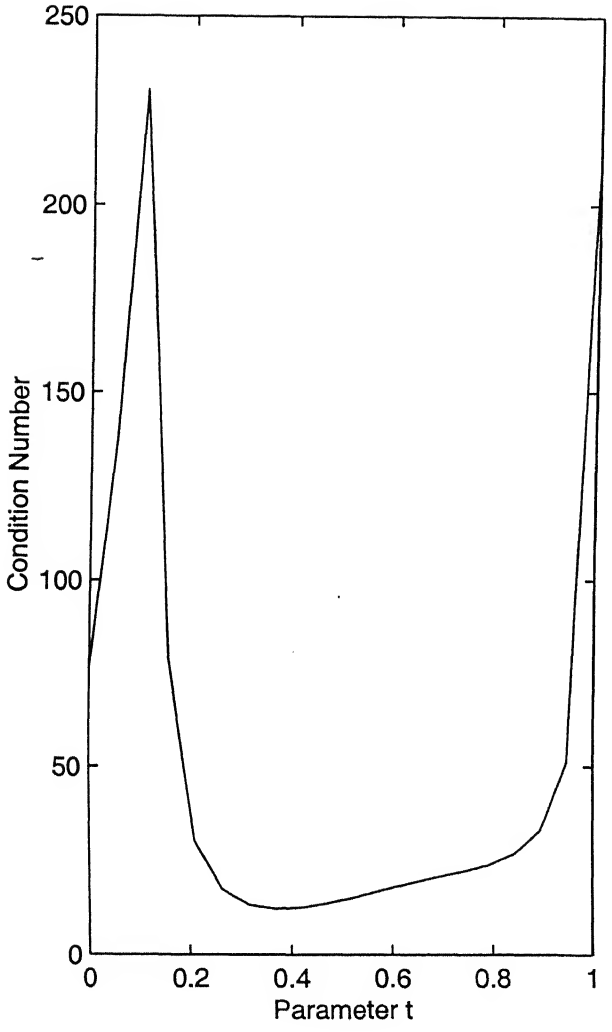


Figure 3.4: Singularity-free planned path for a 8-Bar manipulator.



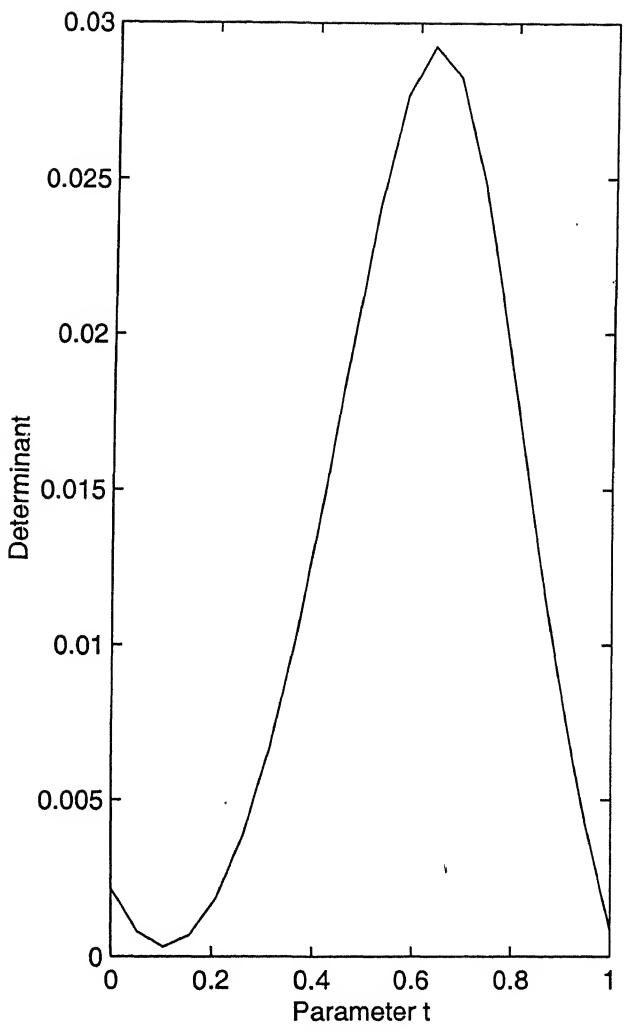


Figure 3.5: Variation of Condition Number and Determinant along the path.

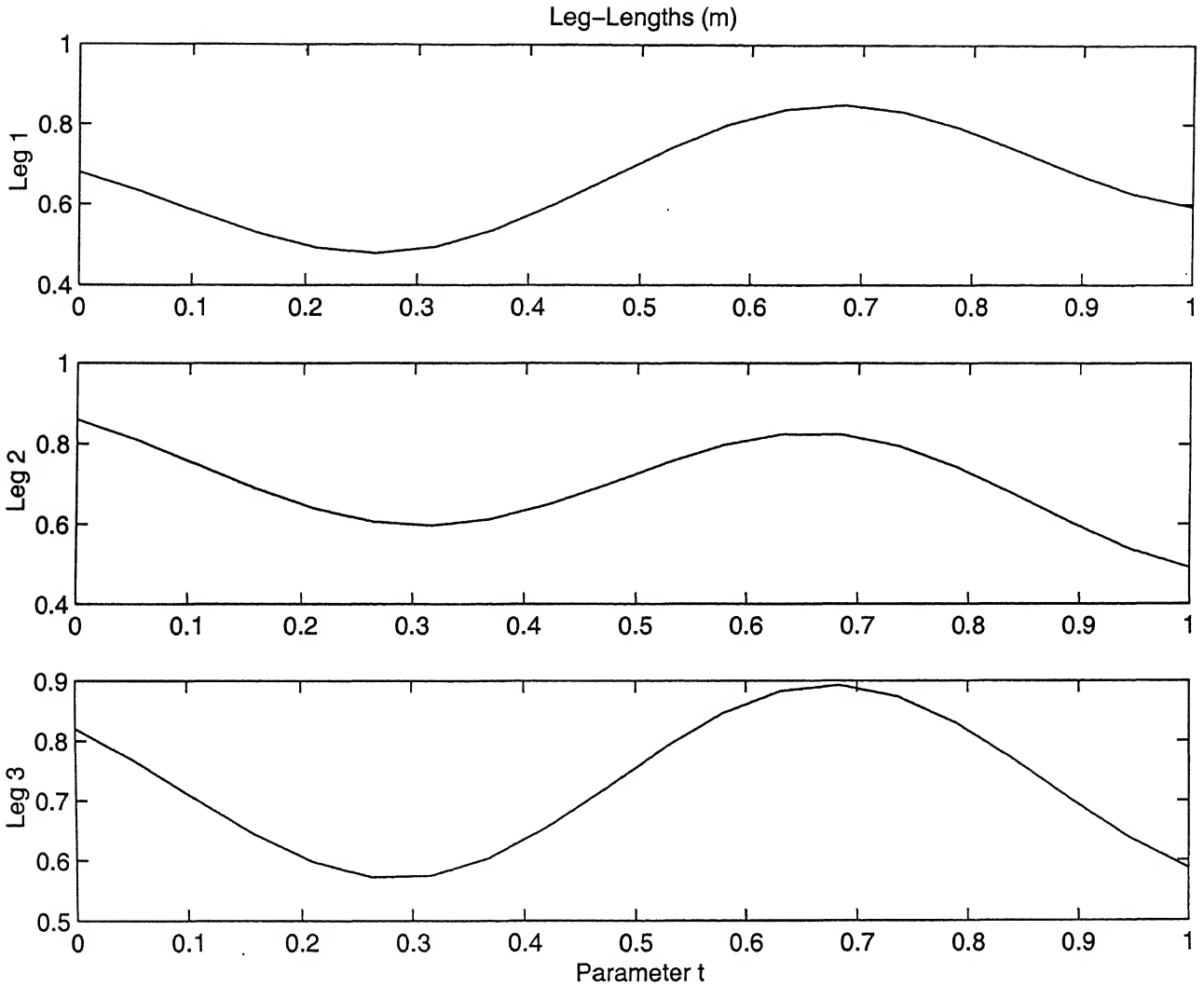


Figure 3.6: Variation of Leg lengths along the path.

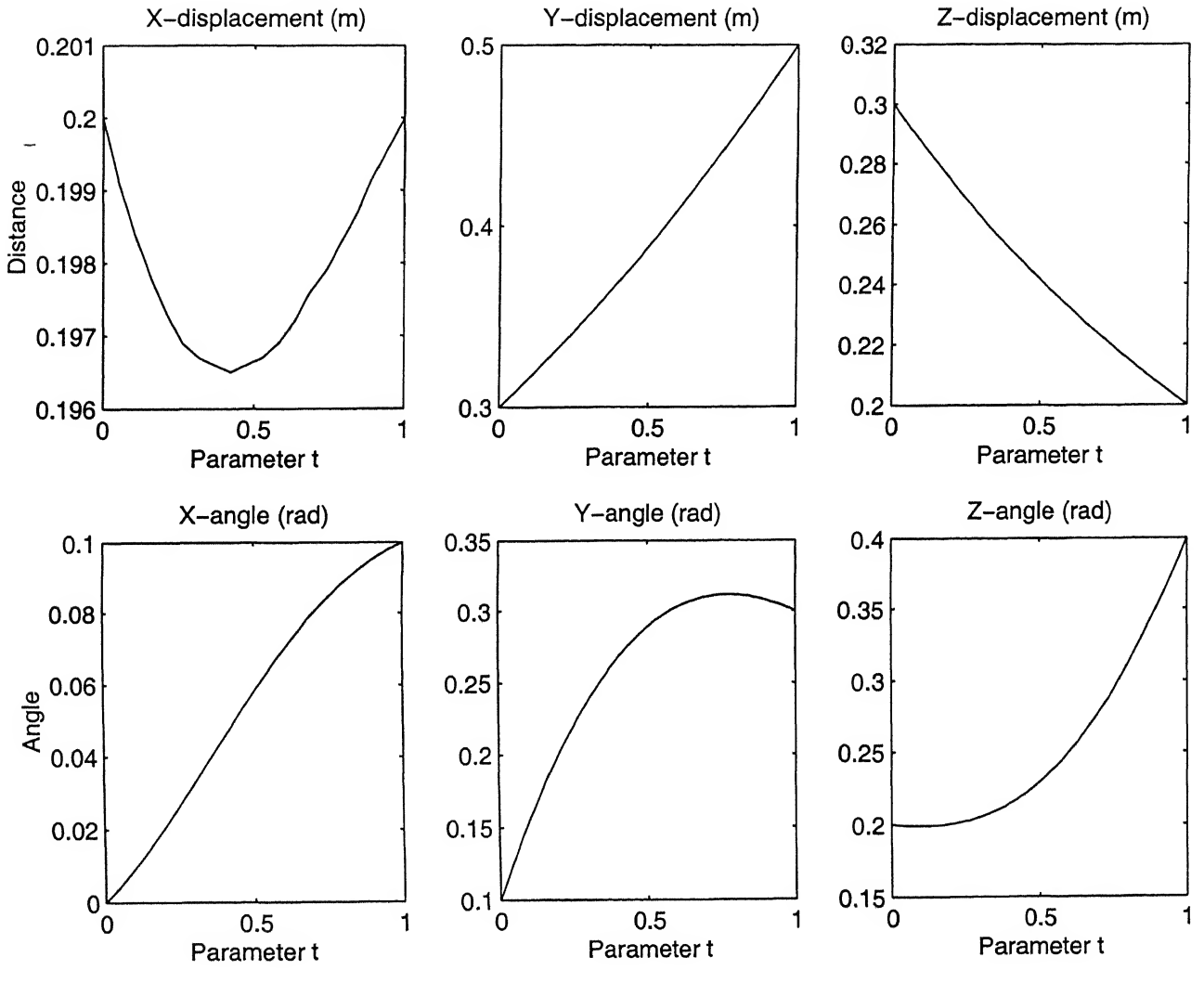


Figure 3.7: Singularity-free planned path for a 3-PRPS Manipulator.

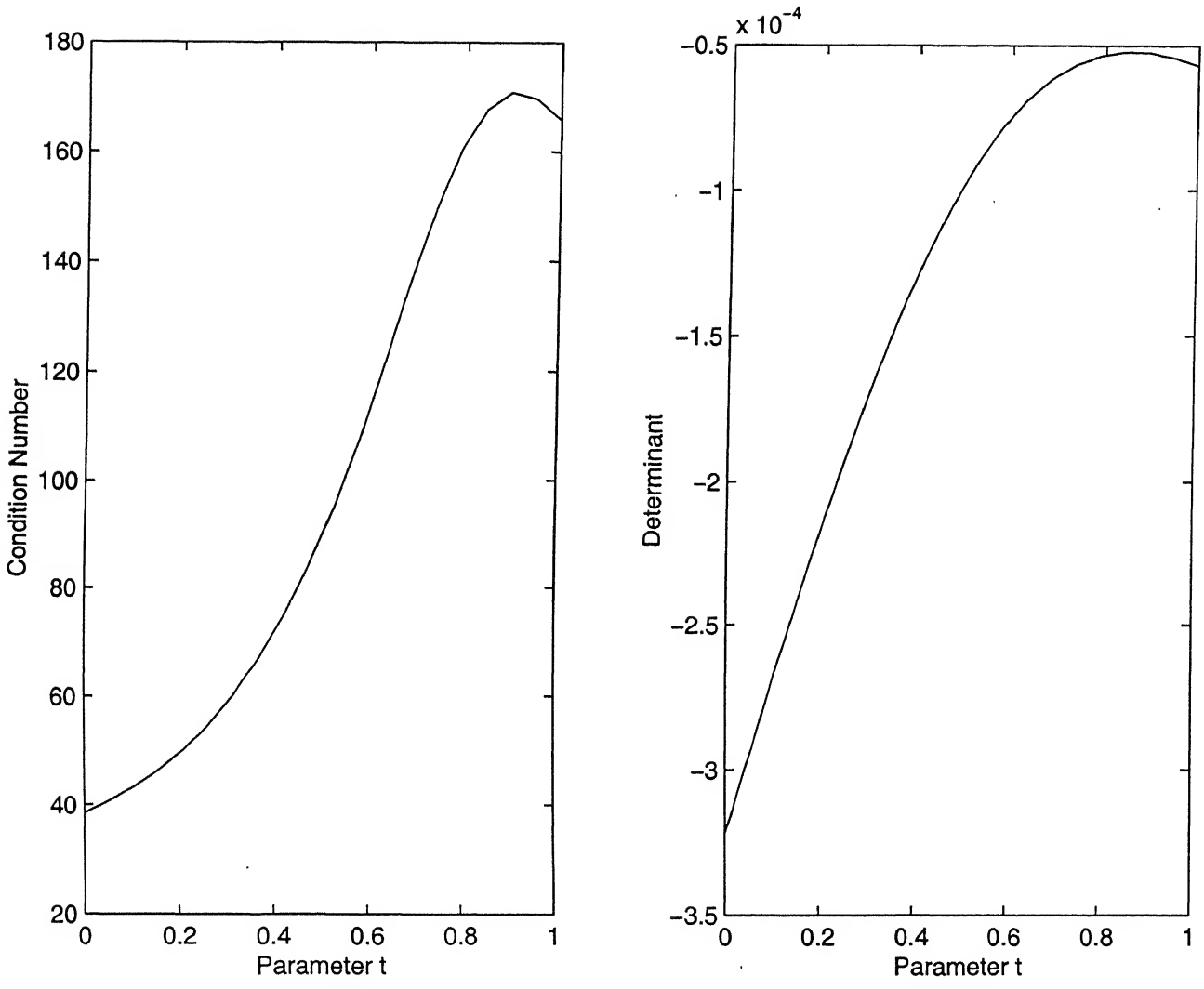


Figure 3.8: Variation of Condition Number and Determinant along the path.

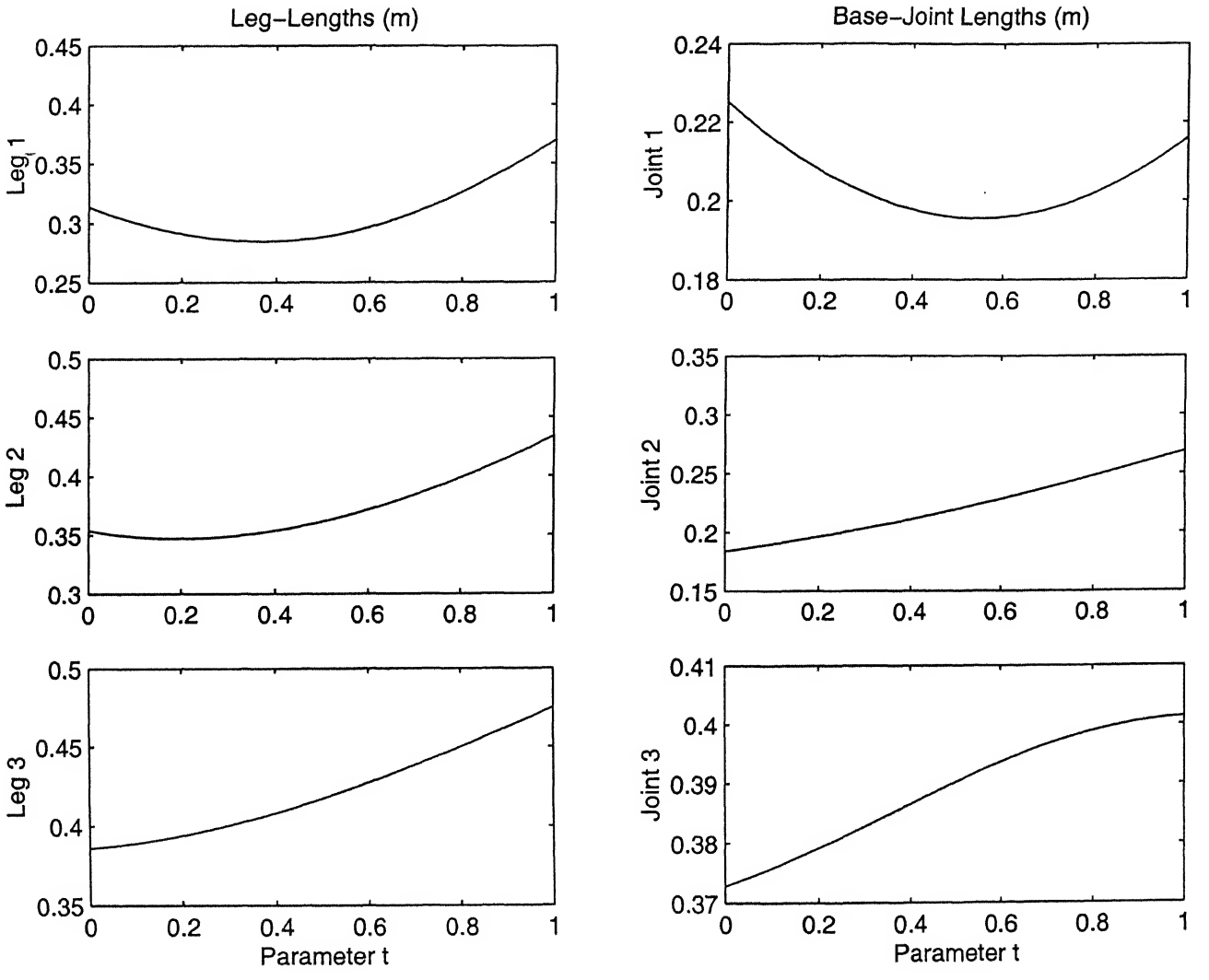


Figure 3.9: Variation of Leg lengths and Base-Joint lengths along the path.

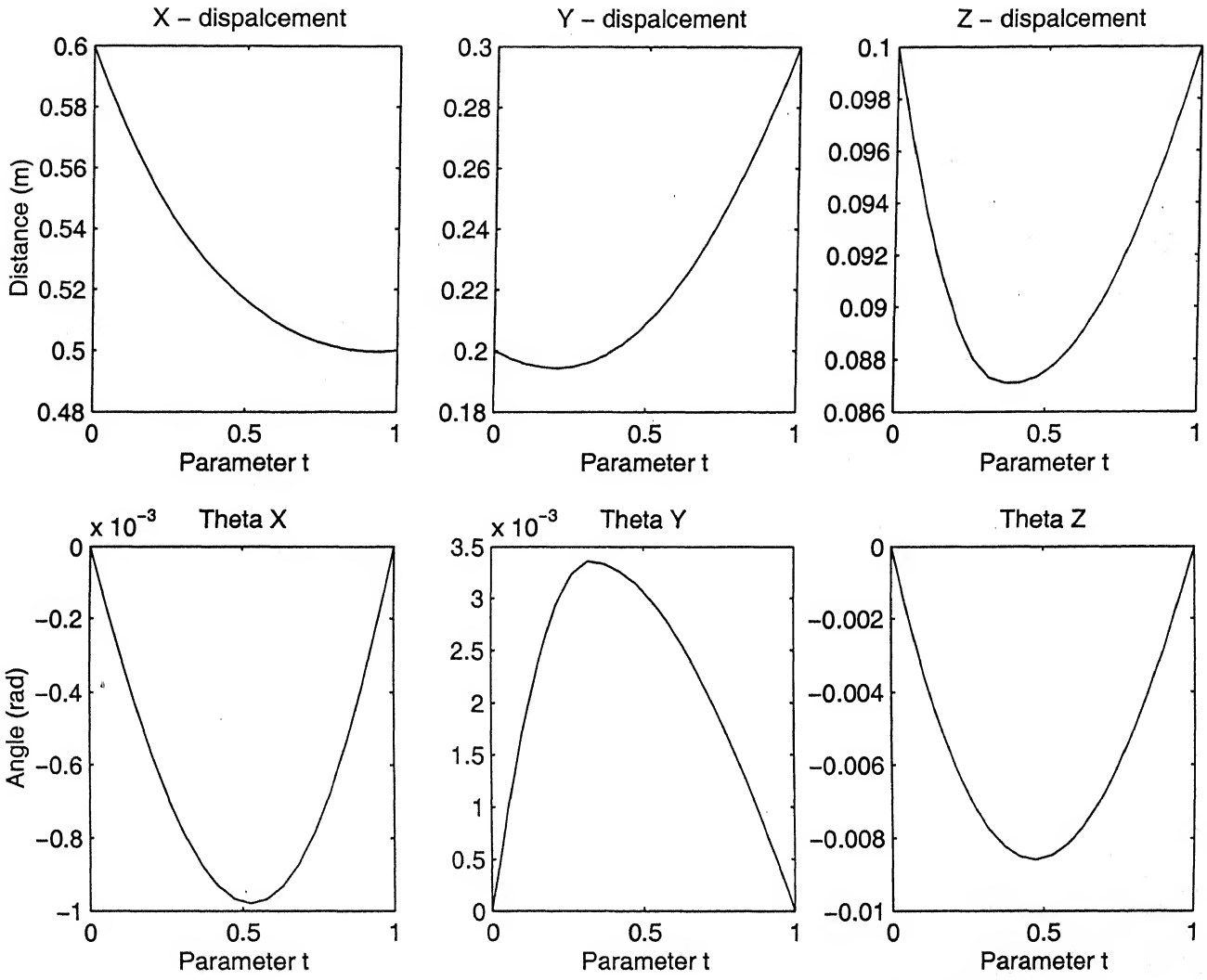


Figure 3.10: Singularity-free planned path for a Stewart platform Manipulator.

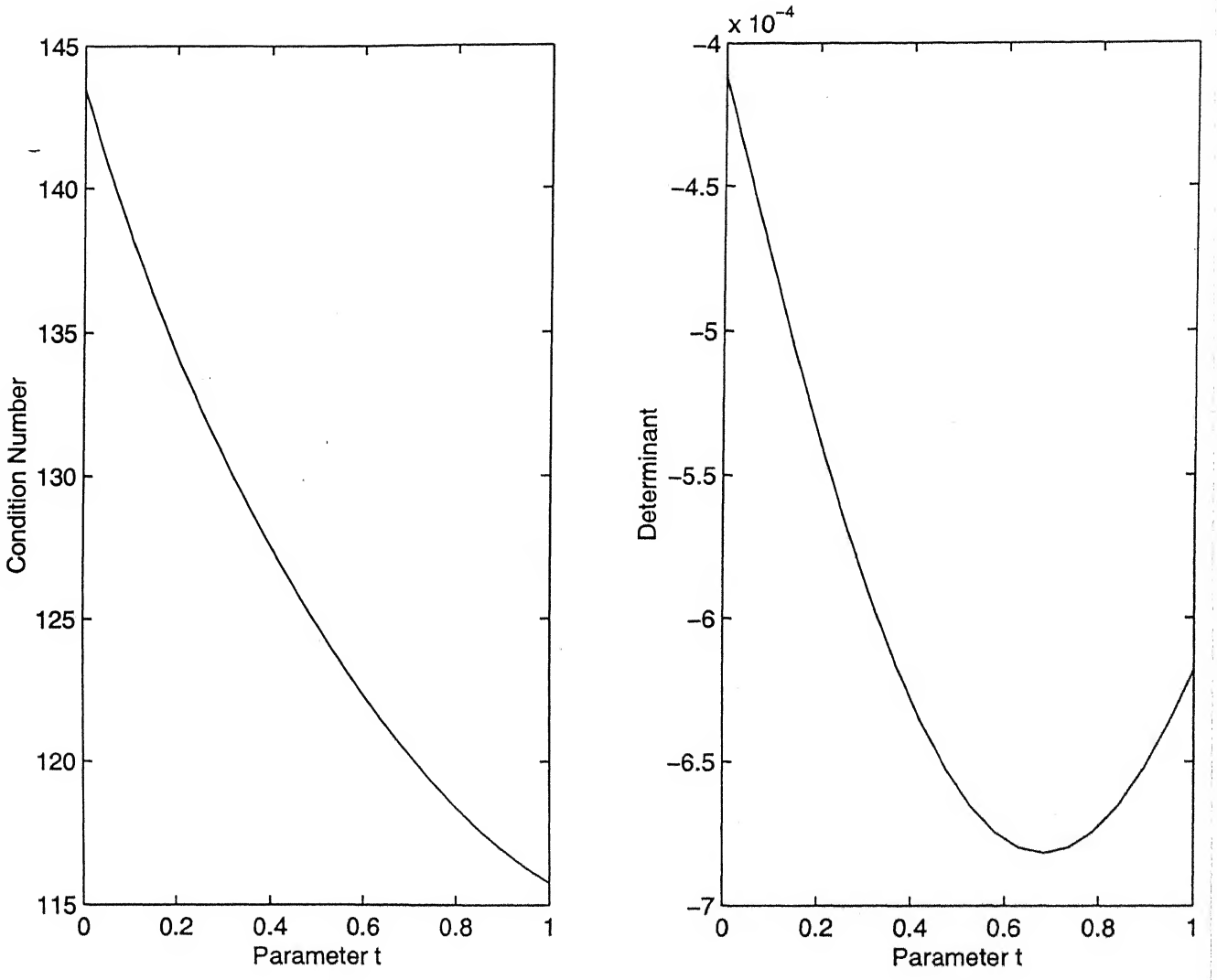


Figure 3.11: Variation of Condition Number and Determinant along the path.

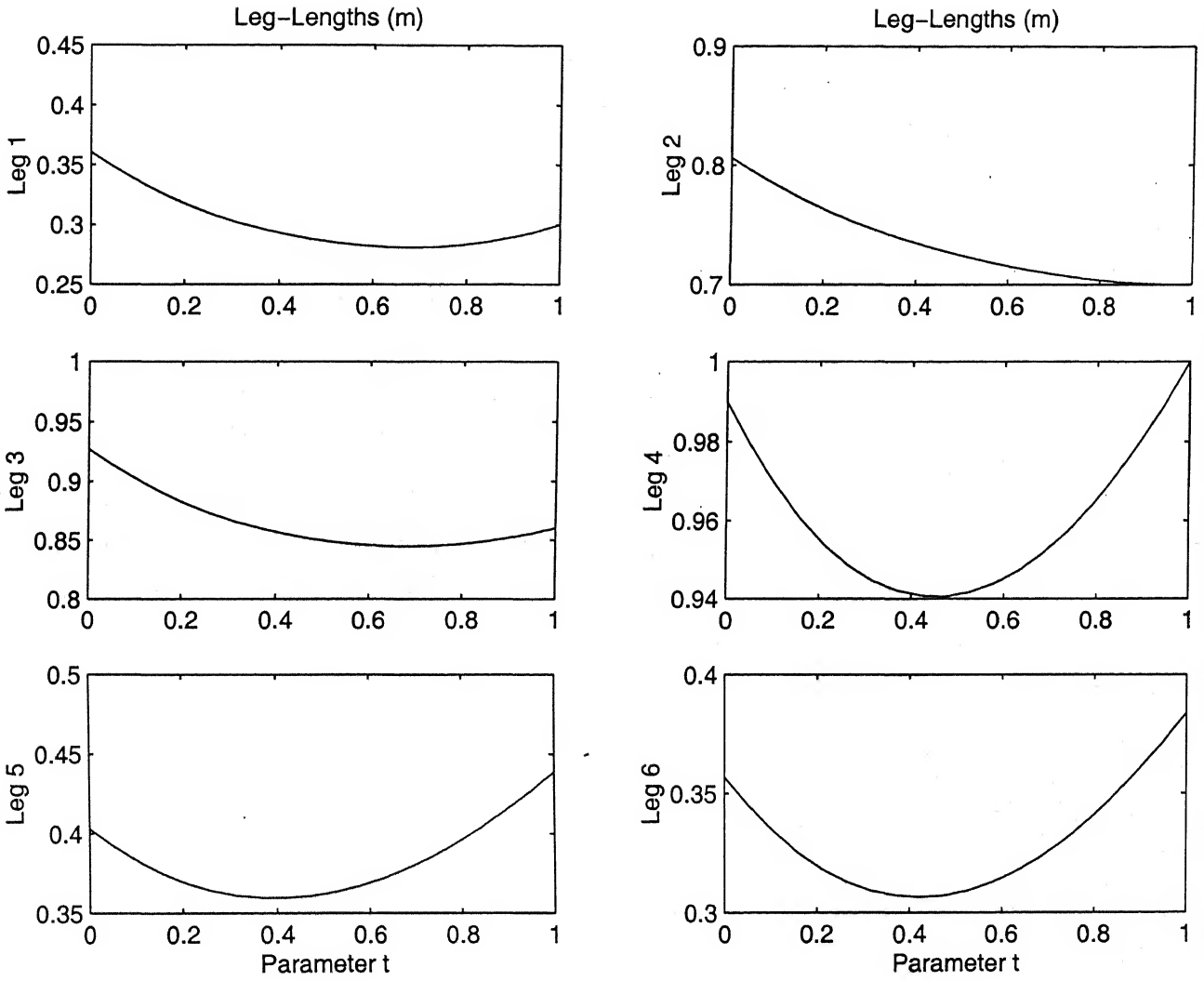


Figure 3.12: Variation of Leg lengths along the path.

Chapter 4

Closure

4.1 Summary

This thesis addresses the problems of control and path-planning of parallel and hybrid manipulators. The methods and concepts used in this thesis are of general applicability for all parallel and hybrid manipulators.

In Chapter 1, parallel and hybrid manipulators have been introduced and their differences with serial manipulators have been highlighted. In the section on Literature Survey, available literature on various aspects of kinematics, dynamics and control, workspaces and singularites, path-planning, and synthesis of parallel manipulators has been presented. Topics have been identified, where the research work has been relatively scarce. Two of these topics have been addressed in the next two chapters.

In Chapter 2, the generalized Newton-Euler Approach is used for deriving the closed-form dynamic equations of parallel and hybrid manipulators. These equations are used for simulations of PD control, Model-based control and Optimal control strategies. The features of each of these strategies are discussed and a comparison of their performance and their computational complexities is presented. Through various numerical examples considered in this Chapter we come to the conclusion that model-based control is the best among all the strategies discussed. Though computationally, it is more expensive than decoupled PD control, its performance is much better than all the other strategies. Moreover, parametric uncertainty can be modelled in model-based control, whereas other strategies require knowledge of exact parameters for good performance.

In Chapter 3, the role of singularities in the mechanics of parallel and hybrid manipulators has been discussed, and the problem of singularity-free path-planning has been addressed. For singularity-free path-planning of these manipulators, the Variational Approach is used for constructing well-conditioned paths for connecting the end-poses. In this approach, a functional is developed as a sum of a kinetic energy term and a potential energy term. The choice of these terms have been discussed, and their effects on the planned paths have been shown through numerical examples. From these examples, we conclude that the Variational Approach is quite

effective for generating singularity-free and well-conditioned paths within the workspaces of th manipulators.

4.2 Suggestions for Future Work

Control and Path-Planning are two fundamental and critical issues related to the developmen of parallel and hybrid manipulators. In this thesis, we have made an attempt to tackle some o the issues related to Control and Path-Planning. The suggestions for future work are as follows

1. Application of the control strategies (considered in this thesis) to more number of paralle manipulators to enhance confidence in the conclusion that model-based control is the best among these.

2. Practical implementation of model-based control strategy for testing its efficacy.

3. Comparison of model-based control with other strategies like adaptive control, and fuzzy control to determine the most effective strategy.

4. To suggest other possible expressions for the functional to be used in the Variational Approach.

Bibliography

[1] W. K. Kim, Y. K. Byun and H. S. Cho, "Closed-Form Forward-Position Solution for a 6-DoF 3-PPSP Parallel Mechanism and Its Implementation", *The International Journal of Robotics Research*, **20**(1), 85-89, 2001.

[2] R. D. Gregario and V. P. Castelli, "Position Analysis in Analytical Form of the 3-PSP Mechanism" *Transactions of the ASME Journal of Mechanical Design*, **123**, 51-58, 2001.

[3] A. K. Dhingra, A. N. Almadi and D. Kohli, "A Grobner-Sylvester Hybrid Method for Closed-Form Displacement Analysis of Mechanisms", *Transactions of the ASME Journal of Mechanical Design* **122**, 431-438, 2000.

[4] Y. Fang and Z. Huang, "Kinematics of a three-degree-of-freedom in-parallel actuated manipulator mechanism, *Mechanism and Machine Theory* **32**(7), 789-796 (1997).

[5] G. R. Dunlop and T. P. Jones, "Position analysis of a 3-dof parallel manipulator", *Mechanism and Machine Theory* **32**(8), 903-920 (1997).

[6] J. M. R. Martinez and J. Duffy, "Forward and Inverse Acceleration Analyses of In-Parallel Manipulators", *Transactions of the ASME Journal of Mechanical Design*, **122**, 299-303, 2000.

[7] L. W. Tsai, "Solving the Inverse Dynamics of a Stewart-Gough Manipulator by the principle of Virtual Work", *Transactions of the ASME Journal of Mechanical Design*, **122**, 3-9, 2000.

[8] B. Dasgupta and T. S. Mruthyunjaya, "A Newton-Euler Formulation for the Inverse Dynamics of the Stewart Platform Manipulator", *Mechanism and Machine Theory* **33**(8), 1135-1152 (1998).

[9] B. Dasgupta and P. Choudhury, "A general method based on the Newton-Euler Approach for the Dynamic Formulation of Parallel Manipulators", *Mechanism and Machine Theory* **34**(6), 801-824 (1999).

[10] D. Basu and A. Ghosal, "Singularity Analysis of Platform-Type Multi-Loop Spatial Mechanisms", *Mechanism and Machine Theory* **32**(3), 375-389 (1997).

[11] F. Gao, X. J. Liu and X. Chen, "The relationships between the shapes of the workspaces and the link-lengths of 3-dof symmetrical planar parallel manipulators", *Mechanism and Machine Theory* **36**, 205-220 (2001).

[12] L. Notash, "Uncertainty Configurations of Parallel Manipulators", *Mechanism and Machine Theory* **33**(1/2), 123-138 (1998).

[13] R. Matone and B. Roth, "In-Parallel Manipulators: A Framework on how to model Actuation Schemes and a study of their effects on singular postures", *Transactions of the ASME Journal of Mechanical Design*, **121**, 2-8, 1999.

[14] J. Wang and C. M. Gosselin, "Kinematic Analysis and Singularity Loci of Spatial four-degree-of-freedom Parallel Manipulators using a Vector Formulation", *Transactions of the ASME Journal of Mechanical Design*, **120**, 555-558, 1998.

[15] R. Ricard and C. M. Gosselin, "On the Determination of the Workspace of Complex Planar Robotic Manipulators", *Transactions of the ASME Journal of Mechanical Design*, **120**, 269-278, 1999.

[16] A. Ghosal and B. Ravani, "A Differential-Geometric Analysis of Singularities of Point Trajectories of Serial and Parallel Manipulators", *Transactions of the ASME Journal of Mechanical Design*, **123**, 80-89, 2001.

[17] I. A. Bonev and J. Ryu, "A geometrical method for computing the constant-orientation workspace of 6-PRRS parallel manipulators", *Mechanism and Machine Theory* **36**, 1-13 (2001).

[18] George H. Pfreundschuh, Vijay Kumar and Thomas G. Sugar, "Design and Control of a 3 DOF in-Parallel Actuated Manipulator",*Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, 1659-1664, April - 1991.

[19] Kevin Cleary and Tatsuo Arai, "A prototype Parallel Manipulator: Kinematics, Construction, Software, Workspace Results, and Singularity Analysis", *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, 566-570, April - 1991.

[20] D. N. Nenchev and M. Uchiyama, "Singularity-consistent path planning and control of parallel robot motion through instantaneous-self-motion type", *IEEE International Conference on Robotics and Automation*, 1864-1870, Minneapolis, 24-26 April, 1996

[21] Bhattacharya, S., Hatwal, H., and Ghosh, A., "Comparison of an exact and an approximate method of singularity avoidance in platform type parallel manipulators", *Mechanism and Machine Theory* **33**(7), 965-974 (1998).

[22] Dasgupta, B., and Mruthyunjaya, T. S., "Singularity-free Path Planning for the Stewart platform manipulator", *Mechanism and Machine Theory* **33**(6), 711-725 (1998).

[23] Innocenti, C., and Castelli, V. P., "Singularity-free evolution from one configuration to another in serial and fully-parallel manipulators", *ASME Journal of Mechanical Design*, 1998, Vol. 120.

[24] X. J. Liu, Z. L. Jin and F. Gao, "Optimum Design of 3-dof spherical parallel manipulators with respect to the conditioning and stiffness indices", *Mechanism and Machine Theory* **35**, 1257-1267 (2000).

[25] J. Lee, J. Duffy and M. Keller, "The Optimum Quality Index for the Stability of In-Parallel Planar Platform Devices", *Transactions of the ASME Journal of Mechanical Design*, **121**, 15-20, 1999.

[26] R. Boudreau and C. M. Gosselin, "The Synthesis of Planar Parallel Manipulators with a Genetic Algorithm", *Transactions of the ASME Journal of Mechanical Design*, **121**, 533-537, 1999.

[27] L. W. Tsai and S. Joshi, "Kinematics and Optimization of a Spatial 3-UPU Parallel Manipulator", *Transactions of the ASME Journal of Mechanical Design*, **122**, 439-446, 2000.

[28] M. R. Driels, U. J. Fan and U. S. Pathre, "The Application of Newton-Euler Recursive Methods to the Derivation of Closed Form Dynamic Equations", *Journal of Robotic Systems* **5**(3), 229-248 (1988).

[29] B. Dasgupta, "The Stewart Platform Manipulator: Dynamic Formulation, Singularity Avoidance and Redundancy", *Ph.D. Thesis, Department of Mechanical Engineering, Indian Institute of Science, Bangalore, India* (1996).

[30] M. Giordano and R. Benea, "On the Dynamics of a 6-RKS Fully-Parallel Robot Manipulator Structure", *Proceedings of the Ninth World Congress on the Theory of Machines and Mechanisms*, 1729-1733 (1995).

[31] B. Dasgupta and T. S. Mruthyunjaya, "Closed-Form Dynamic Equations of the General Stewart Platform through the Newton-Euler Approach", *Mechanism and Machine Theory* 33(7), 993-1012 (1998).

[32] Katsuhiko Ogata, *Modern Control Engineering*, Prentice Hall of India Pvt. Ltd., 1997.

[33] Waldron, K. J. and Hunt, K. H., "Series-parallel dualities in actively coordinated mechanisms", *International Journal of Robotics Research*, 1991, 10(5), 473-480.

[34] Jean-Claude Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, 1991.

[35] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", *International Journal of Robotics Research*, 5(1), 90-98 (1986).

[36] S. J. Buckley, "Fast Motion Planning for Multiple Moving Robots", *Proceedings of the IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, 322-326, 1989.

[37] E. G. Gilbert and D. W. Johnson, "Distance Functions and their Application to Robot Path Planning in the presence of Obstacles", *IEEE Transactions of Robotics and Automation*, RA - 1(1), 21-30, 1985.

[38] Y.K. Hwang and N. Ahuja, "Path Planning Using a Potential Field Representation", *Proceedings of the IEEE International Conference on Robotics and Automation*", Philadelphia, PA, 648-649, 1988.

[39] S. H. Suh and K. G. Shin, "A Variational Dynamic Programming Approach to Robot-Path Planning with a Distance-Safety Criterion", *IEEE Transactions of Robotics and Automation*, 4(3), 334-349, 1988.

[40] Gosselin, C., and Angeles, J., "Singularity Analysis of closed-loop kinematic chains", *IEEE Transactions and Robotic Automation*, 1990,6(3), 281-290.

[41] Ma, O., and Angeles, J., "Architecture singularities of platform manipulators", *Proceedings of the IEEE International Conference of Robotics and Automation*, 1991, pp. 1542-1547.

[42] J.J. Craig, *Introduction to Robotics*, Addison-Wesley Publishing Company Inc., 1986.

Appendix I

Description of the 5-Bar Manipulator of Example I, Chapter 2

Base Points:

$$\mathbf{b} = \begin{bmatrix} 0.0 & 0.2 \\ 0.0 & 0.1 \end{bmatrix} \text{ m}$$

Mass of lower and upper part of each leg:

$$m_d = 3.0 \text{ kg} \qquad \text{and} \qquad m_u = 2.5 \text{ kg}$$

Centres of gravity of lower and upper parts of each leg(in local frame):

$$\mathbf{r}_{d0} = \begin{bmatrix} 0.024 & 0.01 \end{bmatrix}^T \text{ m}$$

$$\mathbf{r}_{u0} = \begin{bmatrix} -0.025 & -0.0 \end{bmatrix}^T \text{ m}$$

Moments of inertia of lower and upper parts of each leg:

$$I_d = 0.01 \text{ kgm}^2 \qquad \text{and} \qquad I_u = 0.04 \text{ kgm}^2$$

Description of the 8-Bar Manipulator of Example II, Chapter 2

Base Points:

$$\mathbf{b} = \begin{bmatrix} 0.0 & 0.8 & 0.5 \\ 0.0 & 0.0 & 0.8 \end{bmatrix} \text{ m}$$

Platform points(in platform frame):

$$\mathbf{p} = \begin{bmatrix} -0.05 & 0.04 & 0.01 \\ -0.05 & -0.05 & 0.05 \end{bmatrix} \text{ m}$$

Mass of lower and upper part of each leg:

$$m_d = 0.5 \text{ kg} \qquad \text{and} \qquad m_u = 0.5 \text{ kg}$$

Centres of gravity of lower and upper parts of each leg(in local frame):

$$\mathbf{r}_{d0} = \begin{bmatrix} 0.2 & 0.005 \end{bmatrix}^T \text{ m}$$

$$\mathbf{r}_{u0} = \begin{bmatrix} 0.248 & 0.0 \end{bmatrix}^T \text{ m}$$

Moments of inertia of lower and upper parts of each leg:

$$I_d = 0.01 \text{ kgm}^2 \qquad \text{and} \qquad I_u = 0.02 \text{ kgm}^2$$

Platform Mass:

$$M_p = 1.0 \text{ kg}$$

Centre of gravity of the platform(in platform frame):

$$\mathbf{R}_0 = \begin{bmatrix} 0.04 & 0.03 \end{bmatrix}^T \text{ m}$$

Moment of Inertia of Platform:

$$I_p = 0.05 \text{ kgm}^2$$

Description of the 3-PRPS Manipulator of Example III, Chapter 2

Base Points:

$$\mathbf{b} = \begin{bmatrix} 0.4 & 0.1 & -0.3 \\ 0.2 & 0.2 & 0.15 \\ 0.0 & 0.1 & 0.1 \end{bmatrix} \text{ m}$$

Platform points(in platform frame):

$$\mathbf{p} = \begin{bmatrix} 0.2 & 0.2 & 0.0 \\ 0.0 & 0.15 & 0.2 \\ 0.1 & 0.0 & 0.0 \end{bmatrix} \text{ m}$$

Unit vectors along the axes of the lower prismatic joints:

$$\mathbf{k} = \begin{bmatrix} -0.6141 & 0.2308 & 0.5535 \\ 0.2714 & 0.4231 & 0.2860 \\ 0.0000 & 0.0077 & 0.0953 \end{bmatrix}$$

Mass of lower, middle and upper part of each leg:

$$m_d = 3.0 \text{ kg} \qquad m_m = 2.0 \text{ kg} \qquad \text{and} \qquad m_u = 1.0 \text{ kg}$$

Centres of gravity of lower and upper parts of each leg(in local frame):

$$\mathbf{r}_{d0} = \begin{bmatrix} 0.14 & -0.08 & 0.0 \end{bmatrix}^T \text{ m}$$

$$\mathbf{r}_{u0} = \begin{bmatrix} -0.18 & 0.08 & 0.0 \end{bmatrix}^T \text{ m}$$

Moments of inertia of lower and upper parts of each leg:

$$\mathbf{I}_{m0} = \begin{bmatrix} 0.010 & 0.005 & 0.007 \\ 0.005 & 0.002 & 0.003 \\ 0.007 & 0.003 & 0.001 \end{bmatrix} \text{ kgm}^2$$

$$\mathbf{I}_{u0} = \begin{bmatrix} 0.005 & 0.002 & 0.002 \\ 0.002 & 0.002 & 0.001 \\ 0.002 & 0.001 & 0.002 \end{bmatrix} \text{ kgm}^2$$

Platform Mass:

$$M_p = 6.0 \text{ kg}$$

Centre of gravity of the platform(in platform frame):

$$\mathbf{R}_0 = \begin{bmatrix} 0.04 & 0.03 & -0.06 \end{bmatrix}^T \text{ m}$$

Moment of Inertia of Platform:

$$\mathbf{I}_{p0} = \begin{bmatrix} 0.020 & 0.001 & 0.003 \\ 0.001 & 0.030 & 0.002 \\ 0.002 & 0.002 & 0.050 \end{bmatrix} \text{ kgm}^2$$

The following matrices $\mathbf{Q}$ and $\mathbf{R}$ have been chosen for the tracking problem

$$\mathbf{Q} = \begin{bmatrix} 2500 & 0 & 0 & 0 \\ 0 & 5000 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R} = 0.00175 * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Choice of Q and R matrices in Example V Chapter 2

The following matrices $\mathbf{Q}$ and $\mathbf{R}$ have been chosen for the tracking problem

$$\mathbf{Q} = \begin{bmatrix} 457 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3200 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R} = 0.0005 * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Appendix III

Description of the 5-Bar Manipulator of Example I, Chapter 3

Base points:

$$[\mathbf{b}_1 \ \ \mathbf{b}_2] = \begin{bmatrix} 0.0 & 0.2 \\ 0.0 & 0.1 \end{bmatrix} \ \text{m}$$

All distances are in metres and angles are in radians.

Description of the 8-Bar Manipulator of Example II, Chapter 3

Base points:

$$[\mathbf{b}_1 \ \ \mathbf{b}_2 \ \ \mathbf{b}_3] = \begin{bmatrix} 0.0 & 0.2 & 0.4 \\ 0.0 & 0.0 & 0.1 \end{bmatrix} \ \text{m}$$

Platform points :

$$[\mathbf{p}_1 \ \ \mathbf{p}_2 \ \ \mathbf{p}_3] = \begin{bmatrix} -0.15 & -0.1 & 0.15 \\ 0.0 & -0.1 & 0.0 \end{bmatrix} \ \text{m}$$

Description of the 3-PRPS Manipulator of Example III, Chapter 3

Base points:

$$[\mathbf{b}_1 \ \mathbf{b}_2 \ \mathbf{b}_3] = \begin{bmatrix} 0.4 & 0.1 & -0.3 \\ 0.2 & 0.2 & 0.15 \\ 0.0 & 0.1 & 0.1 \end{bmatrix} \ \text{m}$$

Platform points :

$$[\mathbf{p}_1 \ \mathbf{p}_2 \ \mathbf{p}_3] = \begin{bmatrix} 0.2 & 0.2 & 0.0 \\ 0.0 & 0.15 & 0.2 \\ 0.1 & 0.0 & 0.0 \end{bmatrix} \ \text{m}$$

The directions of the fixed-axes are :

$$[\mathbf{k}_1 \ \mathbf{k}_2 \ \mathbf{k}_3] = \begin{bmatrix} -0.6141 & 0.2308 & 0.5535 \\ 0.2714 & 0.4231 & 0.2860 \\ 0.5000 & 0.0077 & 0.0953 \end{bmatrix}$$

Description of the Stewart Platform Manipulator of Example IV, Chapter 3

Base points:

$$[\mathbf{b}_1 \ \mathbf{b}_2 \ \mathbf{b}_3 \ \mathbf{b}_4 \ \mathbf{b}_5 \ \mathbf{b}_6] = \begin{bmatrix} 0.6 & 0.1 & -0.3 & -0.3 & 0.2 & 0.5 \\ 0.2 & 0.5 & 0.3 & -0.4 & -0.3 & -0.2 \\ 0.0 & 0.1 & 0.0 & 0.0 & -0.05 & 0.0 \end{bmatrix} \ \text{m}$$

Platform points :

$$[\mathbf{p}_1 \ \mathbf{p}_2 \ \mathbf{p}_3 \ \mathbf{p}_4 \ \mathbf{p}_5 \ \mathbf{p}_6] = \begin{bmatrix} 0.3 & 0.3 & 0.0 & -0.2 & -0.15 & 0.15 \\ 0.0 & 0.2 & 0.3 & 0.1 & -0.2 & -0.15 \\ 0.1 & 0.0 & 0.0 & -0.1 & -0.05 & -0.05 \end{bmatrix} \ \text{m}$$